

**НАЦІОНАЛЬНА АКАДЕМІЯ ПЕДАГОГІЧНИХ НАУК УКРАЇНИ
ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І ЗАСОБІВ НАВЧАННЯ**

*Кваліфікаційна наукова праця
на правах рукопису*

ГОЛОВНЯ ОЛЕНА СЕРГІЇВНА

УДК 004.451.9:(378.4+372.800.2)

ДИСЕРТАЦІЯ

**МЕТОДИКА ЗАСТОСУВАННЯ ТЕХНОЛОГІЙ ВІРТУАЛІЗАЦІЇ
UNIX-ПОДІБНИХ ОПЕРАЦІЙНИХ СИСТЕМ
У ПІДГОТОВЦІ БАКАЛАВРІВ ІНФОРМАТИКИ**

13.00.10 – інформаційно-комунікаційні технології в освіті

01 «Освіта / Педагогіка»

Подається на здобуття наукового ступеня кандидата педагогічних наук.

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.



О. С. Головня

Науковий керівник: **Спірін Олег Михайлович**,
доктор педагогічних наук, професор

Київ – 2019

АНОТАЦІЯ

Головня О. С. Методика застосування технологій віртуалізації Unix-подібних операційних систем у підготовці бакалаврів інформатики. – Кваліфікаційна наукова праця на правах рукопису. Дисертація на здобуття наукового ступеня кандидата педагогічних наук (доктора філософії) за спеціальністю 13.00.10 «Інформаційно-комунікаційні технології в освіті» (01 «Освіта / Педагогіка»). – Інститут інформаційних технологій і засобів навчання НАПН України, Київ, 2019.

Зміст анотації. Дисертацію присвячено вивченню проблеми розроблення навчально-методичного забезпечення для добору та застосування засобів віртуалізації у навчанні Unix-подібних операційних систем (ОС) бакалаврів інформатики. Проаналізовано основні поняття дослідження, сформульовано робоче означення поняття «віртуалізація». Побудовано узагальнену систематизацію технологій віртуалізації. Здійснено аналіз вітчизняного досвіду навчання ОС у ЗВО, під час якого виявлено існування таких проблем, пов'язаних із застосуванням технологій віртуалізації, як апаратні та програмні збої, низька швидкість роботи віртуалізованої ОС, відсутність адаптованих інструкцій, прив'язаність лабораторних занять до певних аудиторій, неоднаковість умов, у яких здійснюється самостійна робота студентів з віртуалізованою ОС. Порівняння балів майбутніх бакалаврів інформатики, одержаних під час проходження курсу з ОС, який передбачав вивчення ОС Linux та Windows поряд із теорією ОС, виявило, що рівень навчальних досягнень, пов'язаних з Linux, був загалом нижчий за рівень навчальних досягнень з дисципліни в цілому. Опитування викладачів ОС у вітчизняних закладах вищої освіти (ЗВО) показало існування низки відмінностей щодо особливостей проведення цього курсу, зокрема різноманітність використовуваних технологій і засобів віртуалізації, досвіду роботи з ними у викладацького складу, оцінювання викладачами факторів добору засобів віртуалізації. Зроблено висновок про необхідність індивідуального добору засобів віртуалізації у курсі з ОС для бакалаврів інформатики, про потребу

залучення альтернативних засобів віртуалізації поряд з основними. Аналіз закордонного досвіду навчання ОС також виявив, зокрема, використання альтернативних засобів віртуалізації.

Уточнено поняття інформатичних компетентностей бакалаврів інформатики щодо ОС. Визначено інформатичні компетентності бакалаврів інформатики щодо ОС: знати та розуміти основні етапи, напрями та тенденції розвитку ОС; орієнтуватися у класифікації ОС, з урахуванням їх архітектури та сфери застосування; знати та розуміти теоретичні основи будови та функціонування ОС; вміти використовувати інтерфейс користувача Unix-подібних ОС (графічний інтерфейс і / або інтерфейс командного рядка); вміти налаштовувати мережне з'єднання у Unix-подібних ОС; вміти встановлювати програмне забезпечення у Unix-подібних ОС, виконувати його наступне оновлення та деінсталяцію; вміти відстежувати роботу процесів та потоків в ОС і керувати ними; вміти здійснювати моніторинг використання різних видів пам'яті; володіти основами роботи з файловою системою у Unix-подібних ОС; володіти основами роботи з налаштуваннями системи безпеки ОС; володіти основами системного програмування мовою C++ у Linux і / або у Windows; вміти одержувати довідкові відомості щодо використання ОС та програмування у них. Визначено педагогічні умови навчання ОС бакалаврів інформатики.

Описано загальну методику дослідження. Побудовано модель застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики, яка складається з цільового, змістового, технологічного, діагностичного й аналітичного компонентів. Модель містить критерії сформованості інформатичних компетентностей бакалаврів інформатики щодо ОС (теоретичний, адміністративний, програмістський, самонавчальний), відповідні показники та рівні (початковий, середній, достатній, високий). Сформульовано критерії добору програмних засобів віртуалізації у навчанні Unix-подібних ОС бакалаврів інформатики (відповідність використовуваної технології віртуалізації потребам курсу; універсальність; ліцензійна чистота; функціональність інтерфейсу; популярність

та підтримка) та відповідні показники. Відзначено суттєву залежність результату оцінювання програмних засобів за пропонованими критеріями і показниками від ваги показників і, як наслідок, підтверджено потребу індивідуального добору засобів віртуалізації у кожному окремому випадку.

Розроблено методику застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики на основі варіативного підходу до використання таких технологій. Підхід полягає у поєднанні кількох засобів віртуалізації у межах одного курсу з ОС задля відповідності індивідуальним особливостям проведення курсу, індивідуальним потребам студентів і забезпечення стійкості у разі програмних та апаратних збоїв. Уточнено процедуру добору засобів віртуалізації для навчання ОС бакалаврів інформатики. Відповідно до запропонованої методики створено методичний посібник з дисципліни «Операційні системи та системне програмування» для навчання майбутніх бакалаврів інформатики та методичні рекомендації «Технології віртуалізації у навчанні операційних систем бакалаврів інформатики» для викладачів ЗВО.

Здійснено експериментальну перевірку методики, наведено результати, здійснено їх узагальнення та інтерпретацію. Для аналізу одержаних даних застосовано статистичні методи, зокрема критерій типу Колмогорова-Смирнова та кутове перетворення Фішера.

Наукова новизна та теоретичне значення одержаних результатів полягають у тому, що вперше: теоретично обґрунтовано та розроблено модель застосування технологій віртуалізації Unix-подібних операційних систем у підготовці бакалаврів інформатики; визначено критерії, показники та рівні сформованості інформатичних компетентностей бакалаврів інформатики щодо операційних систем; уточнено поняття інформатичних компетентностей бакалаврів інформатики щодо операційних систем; критерії добору засобів віртуалізації у навчанні Unix-подібних ОС бакалаврів інформатики; процедуру добору засобів віртуалізації для навчання операційних систем бакалаврів інформатики; дістала подальшого розвитку методика застосування апаратних і

програмних засобів інформатизації освіти у частині, що стосується навчання операційних систем бакалаврів інформатики.

Практичне значення одержаних результатів полягає у тому, що: розроблено методику застосування технологій віртуалізації Unix-подібних операційних систем у підготовці бакалаврів інформатики; розроблено навчально-методичний посібник для бакалаврів інформатики «Операційні системи та системне програмування», що містить лабораторні та самостійні роботи, адаптовані до застосування кількох засобів віртуалізації у процесі навчання операційних систем; розроблено методичні рекомендації «Технології віртуалізації у навчанні операційних систем бакалаврів інформатики» для викладачів ЗВО.

Теоретичні та практичні результати дослідження може бути використано у педагогічних ЗВО викладачами і студентами у процесі навчання Unix-подібних ОС, для добору засобів віртуалізації для навчання ОС, для добору та розробки навчально-методичного забезпечення курсу з ОС.

Ключові слова: технології віртуалізації, засоби віртуалізації, бакалаври інформатики, операційні системи, Unix-подібні операційні системи.

ABSTRACT

Holovnia O. S. Methodology for using Unix-like operating systems virtualization technologies in training bachelors of Informatics. – Qualifying scientific paper, manuscript. The thesis for a Candidate Degree in Pedagogical Sciences on speciality 13.00.10 «Information and Communication Technologies in Education» (01 «Education / Pedagogics»). – Institute of Information Technologies and Learning Tools of NAES of Ukraine, Kyiv, 2019.

Abstract content. The dissertation study investigated the problem of development of training and methodological materials on selecting and using virtualization tools for teaching Unix-like operating systems (OS) to bachelors of Informatics. The basic concepts are analysed, a working definition for the term «virtualization» is provided. The study develops the generalized systematization of

virtualization technologies. Through a research on the experience of teaching OS in Ukrainian higher educational establishments, the dissertation detected a range of issues in using virtualization technologies such as hardware and software failures, low performance of virtualized operating systems, the lack of adapted instructions, dependence on specific computer laboratories, differing conditions of out-of-class work with virtualized OS. Comparison of bachelors' of Informatics scores in OS course, which included studying Linux and Windows OS and also studying the fundamentals of OS theory, showed a generally lower level of learning achievements in Linux than a level of learning achievements in the entire course. A survey of Ukrainian teachers of operating systems course revealed a number of differences in OS courses, including differing experience of using virtualization technologies among the teaching staff, a variety of virtualization technologies applied during a course, differing evaluation of factors for selecting virtualization tools. The research considered that virtualization tools in OS course for bachelors of Informatics need to be selected individually, and alternative virtualization tools need to be used along with main tools. A study on the foreign experience of teaching OS also detected the use of alternative virtualization tools.

The work clarifies the concept of informatics competencies in OS of bachelors of Informatics. The dissertation determines the informatics competencies in OS of bachelors of Informatics, which are as follows: know and understand the main milestones, directions and trends of OS development; describe the OS classifications, based on the architecture and scope of application; know and understand theoretical basis of OS structure and functioning; be able to apply graphical user interface of Unix-like OS (graphical user interface and / or command line interface); be able to do network settings in Unix-like OS; be able to install, update and remove software in Unix-like OS; be able to monitor and manage OS processes and threads; be able to monitor the usage of different memory types; have basic familiarity on manipulating file system in Unix-like OS; have basic familiarity on configuring the OS system security; have basic familiarity on system programming on C++ in Linux and / or Windows; be able to get reference information on OS using and programming. The

work also defines the pedagogical conditions of teaching OS to bachelors of Informatics.

The work outlines the general research methodology. The dissertation specifies the model of using Unix-like OS virtualization technologies in training bachelors of Informatics, containing target, content, technological, diagnostical and analytical components. The model includes the criteria for the determination of the level of the informatics competencies in OS of bachelors of Informatics (theoretical, administrative, programming, self-studying), as well as the corresponding indicators and levels (elementary, basic, intermediate, advanced). The study elaborates the criteria for selecting virtualization software in teaching Unix-like OS (conformity of the underlying virtualization technology with specific features of the course; universality; compliance with the license; interface functionality; popularity and support), as well as the corresponding indicators. The work reveals the sensitivities of the results of software estimation in relation to indicators weighting. These conclusions confirm the need for individual selecting virtualization tools in each particular case.

The dissertation develops the methodology for using Unix-like OS virtualization technologies in training bachelors of Informatics based on the varied approach to applying before-mentioned technologies. The approach involves combining several virtualization tools for meeting individual features of a course, individual students' needs and ensuring a hardware and software fault tolerance. The work specifies the procedure for selecting virtualization tools in teaching OS to bachelors of Informatics. Following the suggested methodology, the study also developed the «Operating Systems and System Programming» handbook for future bachelors of Informatics and methodological guidelines «Virtualization Technologies in Teaching Operating Systems to Bachelors of Informatics» for teachers of higher educational establishments.

The work provides experimental results demonstrating the effectiveness of the elaborated methodology. The analysis of the data received uses statistical methods, including the Kolmogorov-Smirnov test and Fisher's test.

The scientific novelty and theoretical significance of the research are that for the first time: the model of using Unix-like operating systems virtualization technologies in training bachelors of Informatics has been theoretically justified and developed; the criteria for the determination of the level of the informatics competencies in OS of bachelors of Informatics, as well as the corresponding indicators have been specified; the concept of informatics competencies in OS of bachelors of Informatics has been clarified; the criteria for selecting virtualization software in teaching Unix-like OS to bachelors of Informatics have been specified; the procedure for selecting virtualization tools in teaching OS to bachelors of Informatics has been specified; the methodology for using hardware and software tools in the computerization of education in regard to the part concerning teaching the OS course to bachelors of Informatics have got further development.

The practical significance of the results obtained is that the methodology for using Unix-like OS virtualization technologies in training bachelors of Informatics has been developed; the «Operating Systems and System Programming» handbook for bachelors of Informatics containing practical assignments adapted to be used with several virtualization tools has been developed; the methodological guidelines «Virtualization Technologies in Teaching Operating Systems to Bachelors of Informatics» for higher educational establishments teachers has been developed.

The theoretical and practical results of the research can be used in pedagogical higher educational establishments by teachers and students through the process of teaching Unix-like OS, selecting virtualization tools for teaching OS, selecting and development of training and methodological materials on a course in OS.

Keywords: virtualization technologies, virtualization tools, bachelors of Informatics, operating systems, Unix-like operating systems.

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Наукові праці, в яких опубліковано основні наукові результати

Посібники

1. О. С. Головня, *Операційні системи та системне програмування: Методичний посібник для студ. вищ. навч. закл., 2-ге вид., Житомир: Рута, 2016.*

2. О. С. Головня, *Операційні системи та системне програмування: Методичний посібник для студ. вищ. навч. закл., Житомир: Вид-во ЖДУ ім. І. Франка, 2013.*

Методичні рекомендації

3. О. С. Головня, *Технології віртуалізації у навчанні операційних систем бакалаврів інформатики: Методичні рекомендації для викладачів вищ. навч. закл., Житомир: Рута, 2017.*

Статті у наукових фахових виданнях України

4. О. С. Сверчевська, "Репозиторії вільно поширюваного програмного забезпечення в навчанні операційних систем і системного програмування", *Інформаційні технології і засоби навчання*, №24 (4), 2011.

5. О. С. Головня, "Систематизація технологій віртуалізації", *Інформаційні технології в освіті*, №12, с. 127-133, 2012. (включений до міжнародних наукометричних баз)

6. О. С. Головня, "Критерії добору програмних засобів віртуалізації у навчанні Unix-подібних операційних систем", *Інформаційні технології в освіті*, №24, с. 119-133, 2015. (включений до міжнародних наукометричних баз)

7. О. С. Головня, "Аналіз закордонного досвіду навчання операційних систем у вищій школі", *Комп'ютер у школі та сім'ї*, №1, с. 19-23, 2017.

8. О. С. Головня, "Варіативний підхід до застосування засобів віртуалізації Unix-подібних операційних систем у підготовці бакалаврів інформатики", *Науковий часопис НПУ імені М. П. Драгоманова. Серія 2: Комп'ютерно-орієнтовані системи навчання*, №19 (16), с. 228-233, 2017.

9. О. М. Спирін, О. С. Головня, "Застосування технологій віртуалізації

Unix-подібних операційних систем у підготовці бакалаврів інформатики", *Інформаційні технології і засоби навчання*, т. 65, № 3, с. 201-222, 2018. (включений до міжнародних наукометричних баз)

Наукові праці, що засвідчують апробацію матеріалів дисертації

Тези доповідей у наукових виданнях України

10. О. С. Сверчевська, "Вільне програмне забезпечення у вивченні операційної системи Linux", на *VII міжнародній науково-практичній конференції "Інноваційні технології в освіті"*, Ялта, 20-22 вересня, 2010, с. 224-226.

11. О. М. Спірін, О. С. Сверчевська, "Linux та VirtualBox у навчанні абстрактних понять теорії операційних систем", на *FOSS Lviv-2011*, Львів, 1-4 лютого, 2011, с. 149-151.

12. О. С. Головня, "Систематизація технологій віртуалізації", на *звітній науковій конференції Інституту інформаційних технологій і засобів навчання НАПН України*, Київ, 29 березня, 2012, с. 87-88.

13. О. С. Головня, "Психолого-педагогічні особливості студентів напряму підготовки «Інформатика» в контексті навчання операційних систем", на *науково-практичній семінар "Міжнародна співпраця у просторі відкритої освіти"*, Київ, 5 листопада, 2012.

14. О. С. Головня, "Критерії добору віртуалізаційного ПЗ у навчанні Unix-подібних операційних систем", на *IX міжнародній конференції ІТЕА-2014*, Київ, 26 листопада, 2014, с. 124-130.

15. О. С. Головня, "Критерії добору програмних засобів віртуалізації Unix-подібних операційних систем у підготовці бакалаврів інформатики", на *II всеукраїнській науково-практичній конференції молодих науковців "Наукова молодь – 2014"*, Київ, 11 грудня, 2014, с. 101-104.

16. О. С. Головня, "Аналіз досвіду навчання операційних систем у закордонних вищих освітніх закладах", на *Всеукраїнській науково-практичній конференції з міжнародною участю "Сучасні інформаційні технології в освіті"*

та науки", Житомир, 10-11 листопада, 2016, с. 194-199.

17. О. С. Головня, "Поєднання кількох засобів віртуалізації у навчанні бакалаврів інформатики дисципліни «Операційні системи та системне програмування»", на *VII міжнародній науково-технічній конференції "Інформаційно-комп'ютерні технології – 2016"*, Житомир, 22-23 квітня, 2016, с. 220-221.

18. О. С. Головня, "Розробка методичних рекомендацій із застосування технологій віртуалізації у курсі з операційних систем", на *II всеукраїнській науково-практичній конференції з міжнародною участю "Сучасні інформаційні технології в освіті та науці"*, присвяченій 10-й річниці функціонування *Інтернет-порталу E-OLYMP*, Житомир, 9-10 листопада, 2017, с. 119-123.

19. О. С. Головня, "Хмарні інтегровані середовища розробки у курсі з операційних систем для студентів педагогічних спеціальностей: досвід і перспективи", на *III всеукраїнській науково-практичній конференції з міжнародною участю "Сучасні інформаційні технології в освіті та науці"*, Житомир, 8-9 листопада, 2018, с. 327-332.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	14
ВСТУП.....	15
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ЗАСТОСУВАННЯ ТЕХНОЛОГІЙ ВІРТУАЛІЗАЦІЇ UNIX-ПОДІБНИХ ОПЕРАЦІЙНИХ СИСТЕМ У ПІДГОТОВЦІ БАКАЛАВРІВ ІНФОРМАТИКИ	22
1.1. Огляд технологій віртуалізації та перспективи їх використання в освіті.....	22
1.1.1. Поняття віртуалізації.....	22
1.1.2. Систематизація технологій віртуалізації.....	27
1.1.3. Тлумачення віртуалізації в освіті.....	44
1.2. Педагогічні особливості навчання операційних систем бакалаврів інформатики	50
1.2.1. Деякі тенденції сучасної інформатичної освіти	50
1.2.2. Особливості навчання операційних систем у вищій школі.....	53
1.2.3. Педагогічні умови навчання операційних систем	62
1.3. Аналіз закордонного досвіду навчання операційних систем у вищій школі	64
Висновки до розділу 1	78
РОЗДІЛ 2. ПРОЕКТУВАННЯ МЕТОДИКИ ЗАСТОСУВАННЯ ТЕХНОЛОГІЙ ВІРТУАЛІЗАЦІЇ UNIX-ПОДІБНИХ ОПЕРАЦІЙНИХ СИСТЕМ У ПІДГОТОВЦІ БАКАЛАВРІВ ІНФОРМАТИКИ	82
2.1. Загальна методика дослідження.....	82
2.2. Модель застосування технологій віртуалізації Unix-подібних операційних систем у підготовці бакалаврів інформатики	84
2.3. Критерії добору програмних засобів віртуалізації у навчанні бакалаврів інформатики Unix-подібних операційних систем	95
Висновки до розділу 2.....	118
РОЗДІЛ 3. МЕТОДИЧНІ ОСНОВИ ЗАСТОСУВАННЯ ТЕХНОЛОГІЙ ВІРТУАЛІЗАЦІЇ UNIX-ПОДІБНИХ ОПЕРАЦІЙНИХ СИСТЕМ У ПІДГОТОВЦІ БАКАЛАВРІВ ІНФОРМАТИКИ	121
3.1. Мета та завдання застосування технологій віртуалізації Unix-подібних операційних систем у підготовці бакалаврів інформатики	121

3.2. Структура та зміст застосування технологій віртуалізації Unix-подібних операційних систем у підготовці бакалаврів інформатики	123
3.3. Варіативний підхід до застосування засобів віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики	139
3.4. Зміст, форми, методи і засоби навчання під час застосування технологій віртуалізації Unix-подібних операційних систем у межах змістового модуля «Файлові системи»	157
Висновки до розділу 3	174
РОЗДІЛ 4. ОРГАНІЗАЦІЯ, ПРОВЕДЕННЯ ТА РЕЗУЛЬТАТИ	
ПЕДАГОГІЧНОГО ЕКСПЕРИМЕНТУ	176
4.1. Організація педагогічного експерименту	176
4.2. Проведення та аналіз результатів констатувального етапу експерименту	178
4.3. Проведення та аналіз результатів формувального етапу експерименту	204
Висновки до розділу 4	221
ВИСНОВКИ	225
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	229
ДОДАТКИ	259

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ЕГ – експериментальна група

ЗВО – заклад вищої освіти (заклади вищої освіти)

ІКТ – інформаційно-комунікаційні технології

КГ – контрольна група

ОС – операційна система (операційні системи)

ПЗ – програмне забезпечення

ПК – персональний комп'ютер

ЦП – центральний процесор

API – application programming interface – інтерфейс прикладного програмування

CLI – command line interface – інтерфейс командного рядка

GUI – graphical user interface – графічний інтерфейс користувача

IaaS – Infrastructure as a Service – інфраструктура як сервіс

IDE – integrated development environment – інтегроване середовище розробки

IT – informational technologies – інформаційні технології

POSIX – Portable Operating System Interface for Unix – *досл.* портативний стандарт операційних систем для Unix

ВСТУП

Актуальність теми. Навчання операційних систем (ОС), особливостей роботи з ними та основ написання програм для цих ОС є важливою частиною підготовки бакалаврів інформатики. Курс з ОС часто передбачає навчання Unix-подібних ОС у зв'язку з відповідністю означених ОС відкритим стандартам, розповсюдженням Unix-подібних ОС на серверах, поширенням багатьох таких ОС на безкоштовній основі, відкритістю їх програмних кодів тощо.

У процесі практичної роботи з ОС, що вивчаються, часто використовуються технології віртуалізації. Технології віртуалізації імітують різноманітні платформи і дають змогу створювати на їх основі відносно ізольовані середовища. Такі технології застосовують у випадках, коли ОС, що вивчається (наприклад, Linux), відрізняється від ОС, встановленої у комп'ютерних класах (наприклад, Windows), коли треба уникнути надання студентам адміністративних прав, коли студенти працюють на власних комп'ютерах тощо.

Навчання Unix-подібних ОС може здійснюватися у межах низки спеціальностей галузей знань 12 «Інформаційні технології», 11 «Математика та статистика», 01 «Освіта/Педагогіка» та ін., зокрема спеціальності 014.09 «Середня освіта (Інформатика)», яка передбачає підготовку вчителів інформатики.

Підготовці майбутніх учителів інформатики присвячено роботи В. Ю. Бикова [14], А. М. Гуржія [55], Л. В. Брескіної [18], М. І. Жалдака [64], М. П. Лапчика [87], Н. В. Морзе [104; 251], С. М. Прийми [136], Ю. С. Рамського [142], О. М. Спіріна [158], Т. В. Тихонової [171], Ю. В. Триуса [174], Г. Ю. Цибко [182], Г. В. Шугайло [188] та ін. Процес підготовки бакалаврів інформатики вивчали Т. А. Вакалюк [20], Т. Я. Вдовичин [21], Н. Б. Єпик [83], А. П. Кузьменко [83], В. М. Кузьменко [83], У. П. Когут [78] та ін. Навчання інформатичних дисциплін досліджували М. І. Жалдак [63], Т. П. Кобильник [77], В. В. Лаптев [86], М. П. Лапчик [87], Н. В. Морзе [104], Н. І. Рижова [145],

С. О. Семеріков [151], М. В. Швецький [86] та ін.

Теоретичні і практичні аспекти технологій віртуалізації розглянуто у роботах таких авторів, як У. Агесен (O. Agesen) [195], К. Адамс (K. Adams) [195], М. Бен-Єгуда (M. Ben-Yehuda) [204], Р. Голдберг (R. P. Goldberg) [270], Ю. Лі (Y. Li) [244], Р. Наїр (R. Nair) [277], С. Нанда (S. Nanda) [252], Н. Пеннеман (N. Penneman) [269], М. Пірс (M. Pearce) [268], Дж. Попек (G. J. Popok) [270], М. Розенблюм (M. Rosenblum) [273], Дж. Сміт (J. E. Smith) [277], Р. Уліг (R. Uhlig) [284] та ін. Практику використання технологій віртуалізації у навчанні Unix-подібних ОС описують К. Вейл (C. Vaill) [254], Н. Вієно (N. Viennot) [242], О. Лаадан (O. Laadan) [242], Дж. Ні (J. Nieh) [242] та ін.

Освітнє застосування технологій віртуалізації у вигляді хмарних обчислень розглянуто у роботах В. Ю. Бикова [12], С. Г. Литвинової [88], З. С. Сейдаметової [150], М. П. Шишкіної [187], Т. А. Вакалюк [20], О. В. Мерзликіна [97], Ю. Г. Носенко [114], М. В. Попель [133] та ін. Використання технологій віртуалізації для навчання окремих інформатичних дисциплін вивчали А. Є. Батюк, Д. Є. Ванькевич, Г. Г. Злобін [5], Л. В. Павленко, М. П. Павленко [125] та ін. Можливості застосування систем віртуалізації в інформаційно-освітньому просторі закладу освіти досліджено С. М. Яшановим [193]. Застосування хмарних середовищ, що надають засоби навчання для комплексу дисциплін, досліджували О. Г. Глазунова [32], В. В. Лапінський [53], О. П. Горбачевська [53], Г. С. Драган [53], В. П. Олексюк [118], О. В. Якобчук [32] та ін.

Однак недостатньо вивченими лишаються питання добору засобів віртуалізації для навчання низки дисциплін, зокрема ОС, питання добору засобів віртуалізації для підготовки бакалаврів інформатики. Спостерігається відсутність навчально-методичного забезпечення, спрямованого на добір таких засобів та ширше ознайомлення викладачів з технологіями віртуалізації.

Нами виявлено наступні **суперечності**:

- між доцільністю навчання Unix-подібних ОС бакалаврів інформатики і

переважанням комп'ютерних класів з ОС Windows у ЗВО, які здійснюють підготовку бакалаврів інформатики;

- між можливостями застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики і недостатньою розробленістю відповідної методики;

- між різноманітністю сучасних засобів віртуалізації Unix-подібних ОС і нестачею інструктивно-дидактичних матеріалів для викладачів вітчизняних ЗВО з рекомендаціями щодо добору таких засобів.

Проблемою дослідження є необхідність розроблення навчально-методичного забезпечення для добору та застосування засобів віртуалізації у навчанні Unix-подібних операційних систем бакалаврів інформатики. Означена проблема та виявлені суперечності зумовили вибір теми дослідження *«Методика застосування технологій віртуалізації Unix-подібних операційних систем у підготовці бакалаврів інформатики»*.

Зв'язок роботи з науковими програмами, планами, темами. Дисертація виконана в Інституті інформаційних технологій і засобів навчання НАПН України відповідно до тем науково-дослідної роботи «Система науково-організаційного і технологічного забезпечення розвитку мережі електронних бібліотек установ НАПН України» (ДР № 0112U000283); «Система інформаційно-аналітичної підтримки педагогічних досліджень на основі електронних систем відкритого доступу» (ДР № 0115U002234); «Хмарні технології у навчанні майбутніх вчителів інформатики» (НДР №0117U001063) спільної науково-дослідної лабораторії Інституту інформаційних технологій і засобів навчання НАПН України і Житомирського державного університету імені Івана Франка. Тема дисертаційної роботи затверджена на засіданні Вченої ради Інституту інформаційних технологій і засобів навчання НАПН України (протокол №1 від 26.01.2012 р.), узгоджена в Міжвідомчій раді з координації наукових досліджень з педагогічних і психологічних наук в Україні при НАПН України (протокол №7 від 27.09.2011 р.).

Об'єкт дослідження – процес навчання бакалаврів інформатики у ЗВО.

Предмет дослідження – застосування технологій віртуалізації Unix-подібних операційних систем у підготовці бакалаврів інформатики.

Мета дослідження – розробити методику застосування технологій віртуалізації Unix-подібних операційних систем у навчанні бакалаврів інформатики.

Відповідно до мети дослідження поставлено такі **завдання**:

1. З'ясувати стан розробленості проблеми застосування технологій віртуалізації Unix-подібних операційних систем у підготовці бакалаврів інформатики у практиці навчання операційних систем, уточнити понятійний апарат дослідження.

2. Проаналізувати наявні технології з метою добору засобів віртуалізації для навчання Unix-подібних операційних систем у підготовці бакалаврів інформатики.

3. Визначити критерії, показники та рівні сформованості інформатичних компетентностей бакалаврів інформатики щодо операційних систем.

4. Розробити модель застосування технологій віртуалізації Unix-подібних операційних систем у підготовці бакалаврів інформатики.

5. Розробити методику застосування технологій віртуалізації Unix-подібних операційних систем у підготовці бакалаврів інформатики та експериментально перевірити ефективність цієї методики.

Для досягнення поставленої мети й вирішення завдань використано такі **методи дослідження**.

Теоретичні методи: аналіз та узагальнення наукових та науково-методичних джерел з проблеми дослідження, законодавчої та нормативної документації з питань вищої освіти, вітчизняного та закордонного досвіду навчання операційних систем та використання технологій віртуалізації Unix-подібних операційних систем, власного педагогічного досвіду.

Емпіричні методи: анкетування, бесіди з викладачами, студентами, педагогічне спостереження за процесом застосування технологій віртуалізації Unix-подібних операційних систем у підготовці бакалаврів інформатики;

модельовання навчального процесу і педагогічного експерименту в умовах застосування технологій віртуалізації Unix-подібних операційних систем; методи математичної статистики для опрацювання результатів педагогічного експерименту (критерій типу Колмогорова-Смирнова, кутове перетворення Фішера, коефіцієнт узгодженості Кендалла).

Наукова новизна і теоретичне значення одержаних результатів полягає тому, що вперше:

– *теоретично обґрунтовано та розроблено* модель застосування технологій віртуалізації Unix-подібних операційних систем у підготовці бакалаврів інформатики; визначено критерії, показники та рівні сформованості інформатичних компетентностей бакалаврів інформатики щодо операційних систем;

– *уточнено* поняття інформатичних компетентностей бакалаврів інформатики щодо операційних систем; критерії добору засобів віртуалізації у навчанні бакалаврів інформатики Unix-подібних операційних систем; процедуру добору засобів віртуалізації для навчання операційних систем бакалаврів інформатики;

– *дістала подальшого розвитку* методика застосування апаратних і програмних засобів інформатизації освіти у частині, що стосується навчання операційних систем бакалаврів інформатики.

Практичне значення одержаних результатів полягає у тому, що

– *розроблено* методику застосування технологій віртуалізації Unix-подібних операційних систем у підготовці бакалаврів інформатики;

– *розроблено* методичний посібник для бакалаврів інформатики «Операційні системи та системне програмування», що містить лабораторні та самостійні роботи, адаптовані до застосування кількох засобів віртуалізації у процесі навчання операційних систем;

– *розроблено* методичні рекомендації «Технології віртуалізації у навчанні операційних систем бакалаврів інформатики» для викладачів закладів вищої освіти.

Теоретичні та практичні результати дослідження може бути використано у педагогічних закладах вищої освіти викладачами та студентами у процесі навчання Unix-подібних операційних систем, для добору засобів віртуалізації для навчання операційних систем, для добору та розробки навчально-методичного забезпечення курсу з операційних систем.

Упровадження результатів дослідження. Результати дослідження впроваджено в навчальний процес Житомирського державного університету імені Івана Франка (довідка № 587 від 19.06.2017), Сумського державного педагогічного університету імені А. С. Макаренка (довідка № 1151 від 11.05.2018), Дрогобицького державного педагогічного університету імені Івана Франка (довідка № 521 від 17.05.2018), Бердянського державного педагогічного університету (довідка № 57-39/277 від 19.03.2019).

Особистий внесок здобувача. У статті, опублікованій у співавторстві, автору належить: обґрунтування та опис методики застосування технологій віртуалізації Unix-подібних операційних систем у підготовці бакалаврів інформатики, опис результатів експериментальної перевірки пропонованої методики [160].

Апробація результатів дисертації. Основні положення і результати дослідження доповідались та обговорювались на наукових конференціях та семінарах різного рівня: *міжнародних конференціях* – VII Міжнародна науково-практична конференція «Інноваційні технології в освіті» (м. Ялта, 2010 р.); Міжнародна науково-практична конференція FOSS Lviv-2011 (м. Львів, 2011 р.); IX міжнародна конференція "Нові інформаційні технології в освіті для всіх (ІТЕА-2014)" (м. Київ, 2014 р.); VIII Міжнародна науково-технічна конференція «Інформаційно-комп'ютерні технології – 2016» (м. Житомир, 2016 р.); *міжнародних семінарах* – Науково-практичний семінар «Міжнародна співпраця» (м. Київ, 2012 р.); *всеукраїнських конференціях* – II Всеукраїнська науково-практична конференція молодих науковців «Наукова молодь – 2014» (м. Київ, 2014 р.); Всеукраїнська науково-практична конференція з міжнародною участю «Сучасні інформаційні технології в освіті та науці»

(м. Житомир, 2016 р.); II Всеукраїнська науково-практична конференція з міжнародною участю «Сучасні інформаційні технології в освіті та науці», присвячена 10-ій річниці функціонування Інтернет-порталу E-OLYMP (м. Житомир, 2017 р.); III Всеукраїнська науково-практична конференція з міжнародною участю «Сучасні інформаційні технології в освіті та науці» (м. Житомир, 2018 р.); Звітні наукові конференції Інституту інформаційних технологій і засобів навчання НАПН України (м. Київ, 2012-2018 рр.); *всеукраїнських семінарах* – Всеукраїнських методологічних семінарах для молодих учених «Інформаційно-комунікаційні технології в освіті та наукових дослідженнях» (м. Київ, 2011-2016 рр.).

Матеріали і результати дослідження обговорювалися на засіданнях і семінарах кафедри прикладної математики та інформатики Житомирського державного університету імені Івана Франка (2011-2018 рр.), на засіданнях відділу відкритих освітньо-наукових інформаційних систем Інституту інформаційних технологій та засобів навчання НАПН України (2012-2018 рр.).

Публікації. Основні результати дослідження відображено у 19 працях (43,87 д. а., особистий внесок – 43,27 д. а.), серед них 6 статей у наукових фахових виданнях (5,86 д. а., особистий внесок – 5,36 д. а.), 10 тез доповідей у матеріалах конференцій (2,52 д. а., особистий внесок – 2,42 д. а.), 2 методичних посібники (32,65 д. а.), 1 методичні рекомендації (2,84 д. а.).

Структура та обсяг дисертації. Дисертація складається з анотації, списку публікацій здобувача за темою дисертації, переліку умовних позначень, скорочень і термінів, вступу, 4 розділів, висновків, списку використаних джерел і додатків. Повний обсяг дисертації 384 сторінки (14,2 авт. а.), основний текст дисертації викладено на 216 сторінках (8,7 авт. а.). Робота містить 45 таблиць та 65 рисунків, розміщених на 70 сторінках, 20 додатків розміщено на 126 сторінках. Список використаних джерел складає 298 найменувань, серед яких 105 – англійською мовою.

РОЗДІЛ 1.

ТЕОРЕТИЧНІ ОСНОВИ ЗАСТОСУВАННЯ ТЕХНОЛОГІЙ ВІРТУАЛІЗАЦІЇ UNIX-ПОДІБНИХ ОПЕРАЦІЙНИХ СИСТЕМ У ПІДГОТОВЦІ БАКАЛАВРІВ ІНФОРМАТИКИ

1.1. Огляд технологій віртуалізації та перспективи їх використання в освіті

1.1.1. Поняття віртуалізації

Термін «віртуалізація» активно використовується у багатьох галузях знань, зокрема в галузі ІКТ (віртуальні машини, віртуальні контейнери, віртуальні середовища), філософії, політології, психології, соціології (віртуалізація суспільства), економіці (віртуалізація товарів та послуг), освіті (віртуальні навчальні середовища, віртуальні класи, віртуальні спільноти) тощо.

Згідно з англо-українським словником, «віртуальний» («virtual») означає «фактичний, дійсний; можливий, гаданий» [70]. Відповідно до сучасного філософського словника, «віртуальне» тлумачиться як «те, що покладене в надчуттєву сутність і здатне реалізуватися; нематеріальний різновид буття об'єктивних сутностей або суб'єктивних образів» [155].

В. В. Брязкун, проаналізувавши різні підходи до означення віртуального, віртуалізації, віртуальної реальності, пропонує розглядати віртуалізацію як процес переходу від фактичної (актуальної) реальності до віртуальної (потенційної) реальності через певну діяльність людини (енергію) [19].

В галузі ІКТ віртуалізація може тлумачитися у *широкому розумінні* (створення абстракцій для фізичних обчислювальних ресурсів, що лежить в основі роботи будь-якого ПЗ та ОС в цілому) і *вузькому розумінні* (створення додаткових абстракцій з метою організації нових обчислювальних середовищ на базі вже наявних комп'ютерних систем). Розглянемо обидва розуміння віртуалізації детальніше.

У широкому розумінні віртуалізація передбачає створення абстракції для

реальних обчислювальних ресурсів, яку отримують у своє розпорядження користувачі замість приховуваних у такий спосіб реальних ресурсів (оперативної пам'яті, дискових накопичувачів тощо) [189]. Передбачається, що робота з такими абстракціями зручніша за взаємодію з обчислювальними ресурсами напряду, оскільки у такому разі користувач не обов'язково мусить знати деталі роботи з цими ресурсами і позбавляється рутинної праці [168, с. 22-24]. Такі абстракції надають, зокрема, операційні системи (ОС) загалом, мови та середовища програмування, системні бібліотеки, а також низка програмних засобів.

Водночас, віртуалізація в галузі ІКТ може розглядатися і у вузькому розумінні.

Е. Таненбаум описує віртуалізацію як технологію, що «дає можливість одному комп'ютеру стати базою для декількох віртуальних машин», на кожній з яких можуть працювати різні ОС [168]. Подібним чином віртуалізацію означено і у [96]: як засіб організації кількох віртуальних комп'ютерів на одному реальному. У [146] віртуалізацією називають «технологію, яка дозволяє поділити один фізичний сервер на декілька віртуальних, котрі працюють, у певному сенсі, незалежно один від одного». Н. З. Єлманова та С. О. Пахомов під терміном «віртуалізація» мають на увазі можливість одночасного запуску кількох ОС на одному фізичному комп'ютері, тобто, знову ж таки, можливість створення кількох віртуальних комп'ютерів на одному фізичному. При цьому, зауважують автори, ці ОС не обов'язково мають бути однаковими [62]. Аналогічним чином пояснює поняття «віртуалізація» О. К. Гультяєв, називаючи технологію віртуалізації «технологією віртуальних машин». Крім того, О. К. Гультяєв відзначає таку рису віртуалізації, як можливість оперативного переходу з середовища однієї ОС у середовище іншої ОС без перезавантаження комп'ютера [54]. Т. М. Терещенко і К. М. Сафронов означають віртуалізацію як «загальний термін, що охоплює абстракцію ресурсів для багатьох аспектів обчислень» [170]. За А. Сінгхом, віртуалізація – це конструкція або методологія поділу ресурсів комп'ютера на декілька середовищ виконання шляхом

застосування одного або декількох із таких понять та технологій, як програмний та апаратний поділ ресурсів на розділи, режим поділу часу, повна або часткова імітація (англ. *simulation*), емуляція (англ. *emulation*), гарантована якість обслуговування (QoS) тощо [276]. Схожим чином означено віртуалізацію і у [183].

В офіційних описах технології віртуалізації компанії VMware віртуалізацією названо відділення запиту до деякого сервісу від фізичного забезпечення цього сервісу [285]. У [287] віртуалізацією називають технологію приховування фізичних характеристик обчислювальних ресурсів від інших систем, програм чи користувача, котрі взаємодіють з цими ресурсами, відділення подання системи від її реалізації.

Згідно з [237], «віртуалізувати комп'ютер означає змусити комп'ютер здаватися одразу кількома комп'ютерами або цілком іншим комп'ютером». У тій самій публікації знаходимо важливе зауваження: віртуалізацією може вважатися не лише створення ілюзії кількох комп'ютерів на одному, а й навпаки, подання кількох комп'ютерів як єдиного комп'ютера (серверний кластер).

На основі аналізу наведених вище означень було сформульовано робоче означення віртуалізації, також опубліковане у роботі [160]:

Віртуалізація – поняття, що об'єднує технології, засоби, методи тощо, яким притаманні *три головні риси*:

- 1) Поділ ресурсів одного фізичного комп'ютера на декілька взаємно незалежних віртуальних середовищ або, навпаки, об'єднання ресурсів кількох фізичних комп'ютерів в одне віртуальне середовище.
- 2) Оперативність переходу з одного віртуального середовища в інше.
- 3) Приховування реальних фізичних ресурсів та заміна їх абстракціями.

Коли йдеться про технології віртуалізації, у багатьох випадках також вживається термін «віртуальна машина».

Згідно зі статтею Дж. Попека (G. J. Popek) та Р. Голдберга (R. P. Goldberg), віртуальна машина – це «ефективна ізольована копія реальної машини» [270].

За Я. Метлісом (Jan Matlis), віртуальна машина – це «обчислювальне середовище, набір ресурсів і правил роботи якої формується (за допомогою програмного забезпечення) у деякому іншому обчислювальному середовищі» [249].

Б. Сан (Bing Sun) [280] називає віртуальною машиною «віртуальне середовище виконання гостьової операційної системи».

О. К. Гультяєв визначає віртуальну машину як «конкретний екземпляр деякого віртуального обчислювального середовища («віртуального комп'ютера»), створений за допомогою спеціального програмного інструмента». Програмний інструмент, про який ідеться вище, автор називає застосунком віртуальних машин (рос. *приложение виртуальных машин*) [54].

Як бачимо, найчастіше термін «віртуальна машина» використовують на позначення результату віртуалізації, незалежно від того, якими саме поняттями означають цей результат – «копія реальної машини», «обчислювальне середовище», «конкретний екземпляр обчислювального середовища» тощо. Проте у деяких джерелах трапляється й інший підхід: віртуальна машина розглядається не як результат, а як засіб, інструмент віртуалізації. Наприклад, у словнику [23] зазначається: «Віртуальна машина – це програмне забезпечення, що імітує роботу апаратних засобів». Подібний підхід використовується зокрема у [180]. Зауважимо, проте, що пізніше видання згаданого вище словника тлумачить поняття «віртуальна машина» як «неіснуючий абстрактний комп'ютер, роботу якого реалізовано на реальній машині за допомогою програмних засобів» [138, с. 524].

Надалі у даній роботі ми вживатимемо термін «віртуальна машина», коли йтиметься про результат віртуалізації, а не про засіб, за допомогою якого цей результат досягається. Це дасть змогу розрізняти результат і засіб його одержання, а також забезпечить відповідність вживаної термінології більшості публікацій, присвячених питанням віртуалізації. Зазначимо також, що віртуальна машина є поширеним, але далеко не єдиним можливим результатом віртуалізації, про що детальніше йтиметься далі.

Значно менша одностайність спостерігається серед дослідників щодо терміну, який вживається на позначення засобів віртуалізації. Основні терміни, зафіксовані нами, подаємо у таблиці 1.1.

Таблиця 1.1

**Основні терміни, що використовуються для позначення
засобу віртуалізації**

Англійською мовою	Українською мовою
virtual machine application	додаток віртуальних машин, застосування віртуальних машин, застосунки віртуальних машин
virtualization layer	віртуалізаційний рівень, віртуалізаційний прошарок
virtualization software	віртуалізаційне програмне забезпечення, віртуалізаційний софт
virtualization application	додаток віртуалізації, програмний додаток віртуалізації, застосування віртуалізації, застосунок віртуалізації, програмний засіб віртуалізації, засіб віртуалізації
virtual machine monitor	монітор віртуальних машин
hypervisor	гіпервізор
virtualization platform	віртуалізаційна платформа

Для чіткості подальшого викладу мусимо визначитися, які з наведених у таблиці 1.1 термінів надалі використовуватимуться у даній роботі. Проаналізуємо можливі варіанти.

Термін *«додаток віртуальних машин»* досить широко використовується як відповідник англійського терміну «virtual machine application». Проте Термінологічною комісією Держспоживстандарту не рекомендовано перекладати термін «application» як «додаток», натомість запропоновано використовувати такі варіанти перекладу, як «застосування» та «застосунок» [31].

У такому разі одержуємо терміни *«застосування віртуальних машин»* та *«застосунок віртуальних машин»*. Перший термін поза відповідним контекстом легко зрозуміти як напрям використання вже реалізованих віртуальних машин, а це не відповідає змісту даного терміну. Аналогічну ситуацію маємо у випадку

термінів «застосування віртуалізації», «програмне застосування віртуалізації». Тим часом, другий термін – «застосунок віртуальних машин» – містить слово «застосунок», вживання якого стає дедалі розповсюдженішим і яке не викликає непорозумінь, описаних вище. Тому відзначимо термін «застосунок віртуальних машин» як один з можливих для вживання варіантів. Однак треба мати на увазі, що використовувати цей термін правомірно лише стосовно програмних засобів для роботи з віртуальними машинами і неправомірно стосовно засобів для роботи з іншими результатами віртуалізації, зокрема віртуальними контейнерами.

Терміни «гіпервізор», «монітор віртуальних машин» використовується на позначення не всіх засобів віртуалізації, а лише певних їх різновидів, про що йтиметься у п. 1.1.2.

Призначення термінів «віртуалізаційний рівень» та «віртуалізаційний прошарок» вбачаємо у вживанні їх переважно у контексті архітектури тієї чи іншої технології віртуалізації.

Інші терміни буде визначено після детальнішого з'ясування суті та різноманітності технологій віртуалізації далі у цій роботі.

1.1.2. Систематизація технологій віртуалізації

Теоретичні й практичні аспекти технологій віртуалізації досліджено у роботах таких авторів, як У. Агесен (O. Agesen) [195], К. Адамс (K. Adams) [195], М. Бен-Єгуда (M. Ben-Yehuda) [204], М. Бехам (M. Beham) [203], Дж. Бузен (J. P. Buzen) [207], Е. Вітейкер (A. Whitaker) [292], У. Гангліарді (U. O. Gangliardi) [207], Р. Голдберг (R. P. Goldberg) [270], Ю. Лі (Y. Li) [244], Р. Наїр (R. Nair) [277], С. Нанда (S. Nanda) [252], Х. Оу (H. Oh) [255], Н. Пеннеман (N. Penneman) [269], М. Пірс (M. Pearce) [268], Дж. Попек (G. J. Popek) [270], М. Розенблум (M. Rosenblum) [273], Дж. Сміт (J. E. Smith) [277], Р. Уліг (R. Uhlig) [284], Дж. Шугерман (J. Sugerman) [279] та ін.

Терміни «віртуалізація» та «віртуальна машина» виникли у 60-х роках ХХ ст. Упродовж десятиліть технології віртуалізації еволюціонували та

урізноманітнилились, зазначаючи як деяких спадів у розвитку, так і періодів відродження. Історію розвитку технологій віртуалізації розглянуто у додатку А.

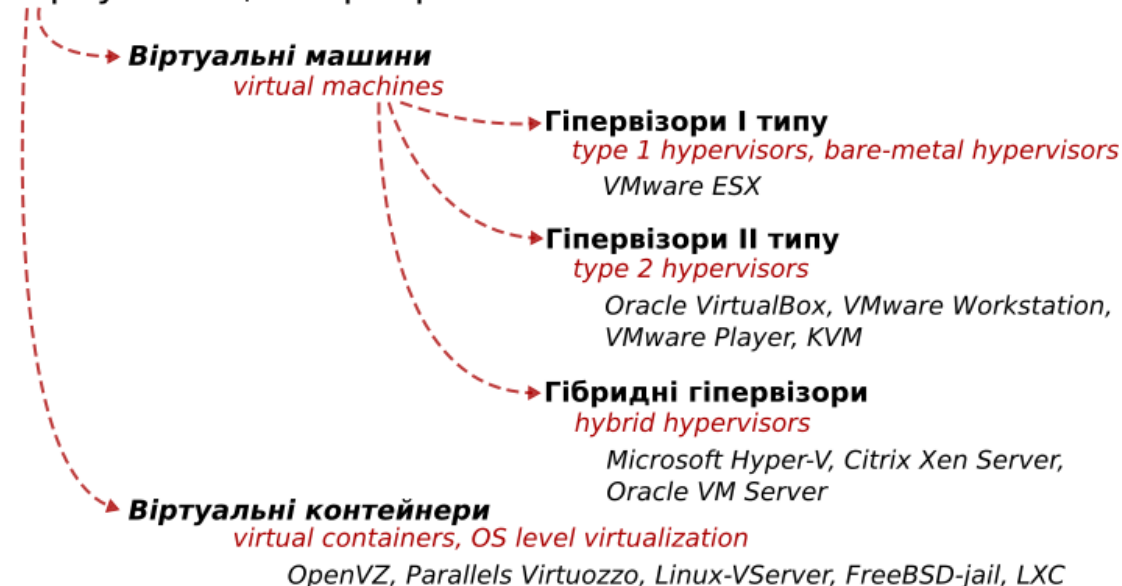
Розуміння основ технологій віртуалізації та відмінностей між їх різновидами суттєво утруднюються значним розмаїттям цих технологій і, як наслідок, відсутністю в опрацьованих джерелах єдиного підходу до систематизації технологій віртуалізації. Проаналізовано систематизації технологій віртуалізації Е. Таненбаума (Andrew Tanenbaum) [168], Н. З. Єлманової та С. О. Пахомова [62], Я. Метліса (Jan Matlis) [249], Б. Сана (Sun Bing) [280], О. К. Гульятєва [54], М. Т. Джонса (M. T. Jones) [237], а також систематизацію, запропоновану у публікації [4], та згодом уточнено та доповнено, враховуючи варіанти систематизації, наведені у роботах авторів М. Розенблум (Mendel Rosenblum) [273], С. Нанда (Susanta Nanda), З. Чиуе (Tzi-cker Chiueh) [252], Дж. Сміт (James E. Smith), Р. Наїр (Ravi Nair) [277], Ю. Лі (Yunfa Li), В. Лі (Wanqing Li), К. Цзян (Congfeng Jiang) [244], С. М. Яшанов [193]. Опис проаналізованих систематизацій наведено у додатку Б.

На основі розглянутих варіантів систематизації було вироблено *узагальнену систематизацію технологій віртуалізації*. Технології віртуалізації систематизовано за напрямом (*що віртуалізуємо?*) та за методом (*як віртуалізуємо?*). За *напрямом віртуалізації* виокремлюємо: віртуалізацію серверів, віртуалізацію настільних ОС, віртуалізацію програмних застосунків та віртуалізацію подань (рис. 1.1).

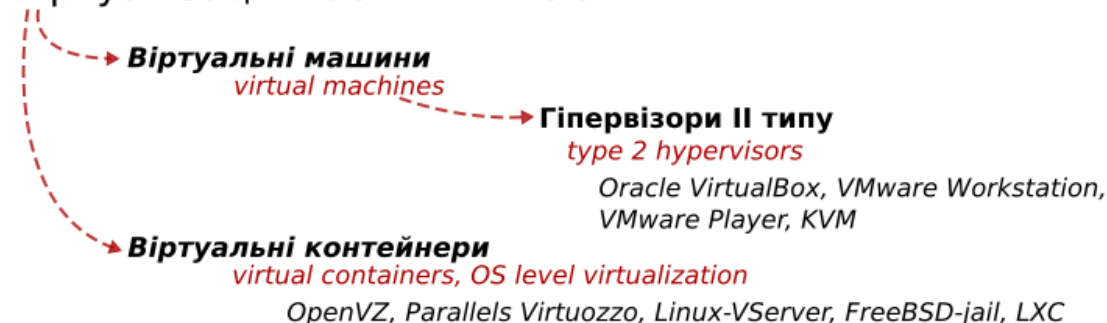
Віртуалізація серверів (server virtualization), відповідно до своєї назви, передбачає, що віртуалізується сервер (веб-сервер, поштовий сервер, файловий сервер, сервер для спеціалізованого ПЗ тощо) [62]. Водночас, для комп'ютера, на якому фізично працює такий сервер, роль апаратної бази для сервера не обов'язково є основною. Машина може використовуватися передусім як настільний ПК, з якого час від часу також запускається віртуальний сервер (зокрема, з навчальною метою). В інших випадках комп'ютер, навпаки, виділено для функціонування передусім як сервер. Від описаних вище обставин залежить вибір технологій та засобів віртуалізації [278, с. 596].

Систематизація технологій віртуалізації за напрямом

Віртуалізація серверів



Віртуалізація настільних ОС



Віртуалізація програмних застосунків

application virtualization

Citrix XenApp, VMware ThinApp, Microsoft App-V

Віртуалізація робочого столу

desktop virtualization

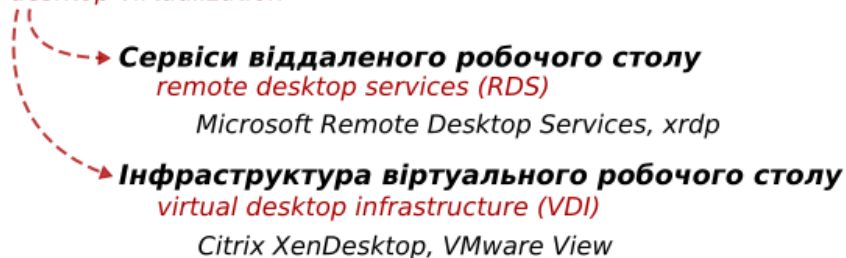


Рис. 1.1. Узагальнена систематизація технологій віртуалізації за напрямом

Загалом, віртуалізація серверів здійснюється на основі *віртуальних машин, віртуальних контейнерів* чи поєднання цих технологій.

Технологія **віртуальних машин** (virtual machines) передбачає імітацію у межах реального комп'ютера одного або кількох віртуальних комп'ютерів, на кожний з яких може бути встановлено окрему ОС з окремим ядром. Кожна віртуальна машина використовує ресурси реальної машини (основну пам'ять, час ЦП, дисковий простір, мережні пристрої тощо). Так, віртуальній машині виділяється деякий обсяг основної пам'яті від загального обсягу основної пам'яті фізичного комп'ютера, віртуальний жорсткий диск, який є файлом спеціального формату на реальному жорсткому диску.

Здійснює розподіл ресурсів між віртуальними машинами та організовує взаємодію віртуальних машин з фізичним обладнанням спеціальне віртуалізаційне ПЗ – **гіпервізор** (hypervisor). Крім того, гіпервізор надає користувачеві інтерфейс для створення віртуальних машин та керування ними [169].

Наполягаємо на необхідності розмежування термінів «гіпервізор» та «віртуальна машина» й уживання першого для позначення засобу віртуалізації, а другого – для позначення результату віртуалізації (віртуальну машину створено і запущено засобами гіпервізора).

Розрізняють гіпервізори I типу, гіпервізори II типу та гібридні гіпервізори.

Гіпервізори I типу (type 1 hypervisors) також називають гіпервізорами, «виконуваними на голому залізі» (bare-metal hypervisors). Гіпервізор I типу не має під собою ОС і сам є мінімальною ОС. Прикладом гіпервізора I типу є VMware ESXi [223].

Гіпервізори II типу (type 2 hypervisors) функціонують як ПЗ у межах попередньо встановленої на фізичний комп'ютер ОС (*основної ОС*). За допомогою гіпервізора II типу всередині основної ОС запускають одну чи декілька віртуальних машин, кожна – з власною ОС. ОС, яка працює на віртуальній машині, називають *гостьовою ОС* [169].

Прикладами гіпервізорів II типу є Oracle VirtualBox, VMware Player, VMware Workstation, KVM та ін.

Також для позначення передусім гіпервізорів II типу використовується згаданий раніше термін *«монітори віртуальних машин»* (virtual machine monitors). Водночас, історично даний термін вживався для позначення всіх гіпервізорів, і таке його застосування має місце дотепер, тому надалі у цій роботі замість терміну *«монітори віртуальних машин»* послуговуватимемося терміном *«гіпервізор II типу»*.

Гібридні гіпервізори (hybrid hypervisors) є поєднанням гіпервізорів I та II типів [62]. Технологія гібридних гіпервізорів передбачає наявність, з одного боку, гіпервізора I типу, що працює безпосередньо на апаратному забезпеченні, а з іншого – спеціальної сервісної ОС, котра функціонує під управлінням гіпервізора I типу. На практиці гібридні гіпервізори ближчі до гіпервізорів I типу і не завжди виділяються в окремий клас. Гібридними можна вважати, зокрема, гіпервізори Microsoft Hyper-V, Citrix Xen Server, Oracle VM Server.

Технологія **віртуальних контейнерів** (virtual containers), або віртуалізація рівня ОС (operating system level virtualization), передбачає створення і паралельну роботу у межах одного фізичного комп'ютера окремих віртуальних середовищ (контейнерів). Контейнери відносно незалежні, проте використовують спільне ядро ОС. Завдяки спільному ядру віртуальні контейнери є загалом швидшими за віртуальні машини, та водночас це унеможливорює запуск у контейнерах ОС на основі різних ядер. Технологія віртуальних контейнерів застосовна лише для Unix-подібних ОС [235; 252].

Прикладами реалізації технологій віртуальних контейнерів є OpenVZ, Parallels Virtuozzo, Linux-VServer, FreeBSD-jail, LXC, Canonical LXD та ін.

У зв'язку з особливостями своєї архітектури, гіпервізори II типу використовуються для віртуалізації серверів рідше за інші типи гіпервізорів та віртуальні контейнери. Оскільки гіпервізори II типу встановлюються поверх основної ОС, значний обсяг ресурсів витрачається на роботу цієї ОС, й відповідно, менший обсяг ресурсів лишається для роботи віртуальних машин

[278, с. 596]. Сервери на основі гіпервізорів II типу частіше обирають для пробної та тимчасової роботи, створюються з навчальною метою тощо. У таких випадках гіпервізори II типу, навпаки, можуть мати переваги. Зокрема, цей тип гіпервізорів потребує порівняно невеликих організаційних зусиль у створенні та видаленні, особливо якщо гіпервізор вже встановлено і використовується з іншою метою (наприклад, вивчення ОС, відмінної від основної).

Віртуалізація настільних ОС (local desktop virtualization) переважно передбачає застосування гіпервізорів II типу або віртуальних контейнерів. Така віртуалізація не має на меті створення віртуального сервера, натомість може мати на меті цілу низку інших застосувань, таких як розробка і налагодження ПЗ для різних ОС (різні дистрибутиви, версії, випуски однієї ОС), запуск застосунків однієї ОС на комп'ютері з іншою ОС, навчання ОС тощо. Зауважимо, що в окремих випадках ці та інші завдання реалізуються й іншими засобами віртуалізації, у тому числі гіпервізорів I типу та гібридних гіпервізорів, про що ще йтиметься далі.

Важливо розуміти, що англomовний відповідник віртуалізації настільних ОС – «local desktop virtualization» – потребує обережного використання. Словосполучення «desktop virtualization» вживається на позначення іншого напрямку віртуалізації – віртуалізації робочого столу. Термін «віртуалізація настільних ОС» вживається лише деякими дослідниками, зокрема [282]. Водночас, у розробленій узагальненій систематизації технологій віртуалізації даний напрям збережено, оскільки під час використання віртуальних машин у навчанні ОС не обов'язково йдеться про віртуалізацію серверів. Віртуалізація серверів та віртуалізація настільних ОС може бути об'єднана в один напрям – *віртуалізація платформ*. Англomовними відповідниками є «platform virtualization» [229], а також «hardware-level virtualization» [273], «hardware virtualization» [252], причому останній не є синонімічним до «hardware-assisted virtualization» [244], а тому не може бути перекладений як «апаратна віртуалізація». Як бачимо, така заміна, на жаль, не вирішує проблему неоднозначності термінології.

У ході *віртуалізації програмних застосунків* (application virtualization) відбувається створення віртуального середовища для кожного екземпляра програми [199]. Це дає змогу застосунку працювати у середовищах, які цей застосунок початково не підтримують. Віртуалізація програмних застосунків суттєво полегшує міграцію застосунків (наприклад, портативні версії програм). При цьому віртуалізація програмних застосунків потребує менше ресурсів, ніж віртуалізація всієї ОС.

Технологія віртуалізації програмних застосунків використовується у Citrix XenApp [294], VMware ThinApp [281], Microsoft App-V та ін.

Віртуалізація робочого столу (desktop virtualization) є віртуалізацією середовища робочого столу, включаючи застосунки користувача. Віртуалізація робочого столу може передбачати як віддалену взаємодію з обчислювальними потужностями, за якої процеси користувача виконуються не на його пристрої (ПК, ноутбук, тонкому клієнті, планшеті, смартфоні тощо), а на сервері, так і локальний запуск програм користувача й локальне зберігання його даних. Часто обидва варіанти взаємодії поєднують [297].

Віртуалізація робочого столу здійснюється на основі двох базових підходів: сервісів віддаленого робочого столу та інфраструктури віддаленого робочого столу.

Сервіси віддаленого робочого столу (remote desktop services, RDS) ґрунтуються на наданні спільного доступу до ОС серверу, причому йдеться про один екземпляр цієї ОС для всіх користувачів [296].

Прикладами реалізації технології сервісів віддаленого робочого столу є Microsoft Remote Desktop Services, xrdp.

Інфраструктура віддаленого робочого столу (virtual desktop infrastructure, VDI) забезпечує кожному користувачу доступ до окремого екземпляра ОС [294]. Технологію VDI втілено, зокрема, у Citrix XenDesktop, VMware View.

За методом віртуалізації виокремимо, передусім, програмну та апаратну віртуалізацію (рис. 1.2).

Систематизація технологій віртуалізації за методом

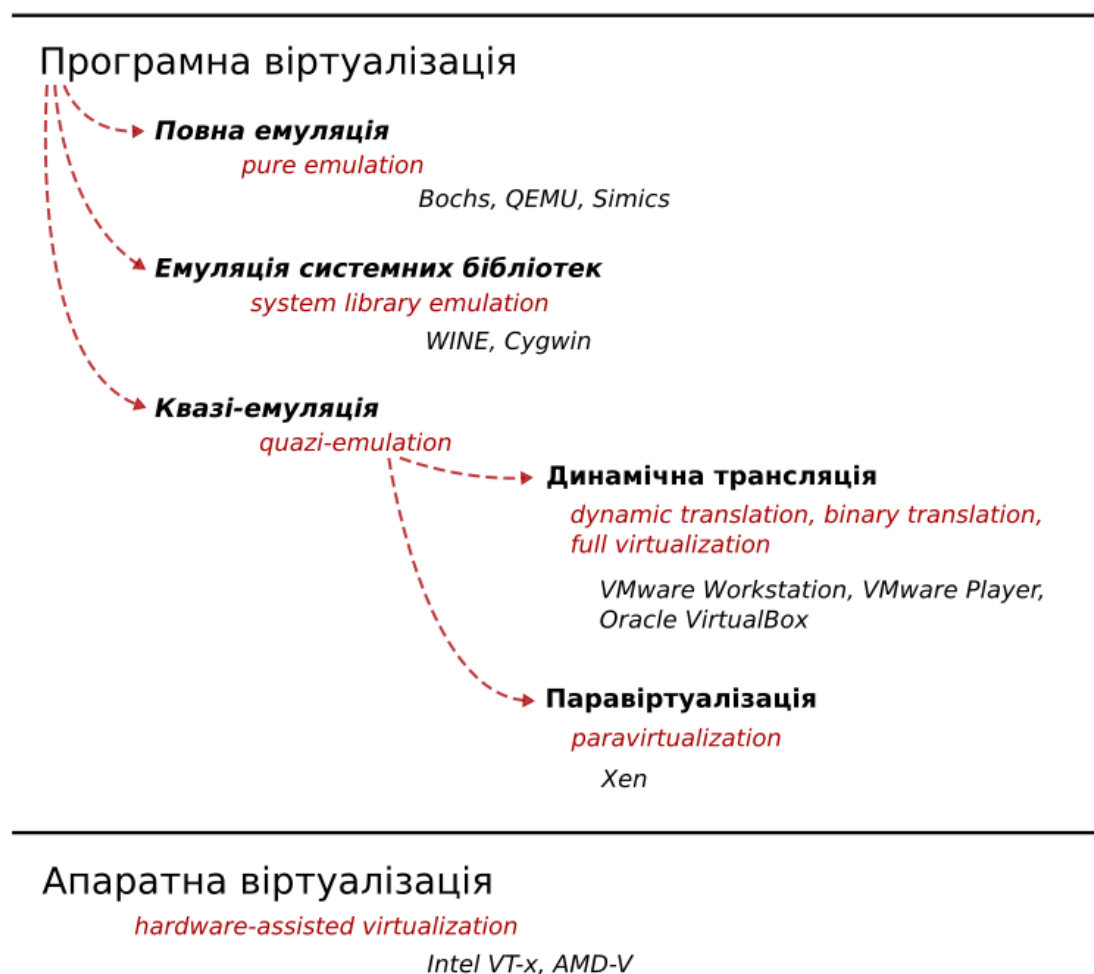


Рис. 1.2. Узагальнена систематизація технологій віртуалізації за методом

Програмну віртуалізацію (software-assisted virtualization) можна поділити на повну емуляцію, емуляцію системних бібліотек та квазіемуляцію.

Метод повної емуляції (pure emulators) передбачає імітацію всього ПЗ комп'ютера, включно з процесором. Продуктивність повної емуляції, порівняно з пізнішими технологіями, є порівняно невисокою, натомість вона дозволяє зімітувати машину однієї архітектури на машині іншої архітектури [278, с. 596; 169]. Прикладами реалізації повного емулятора є програмні засоби Bochs, QEMU, Simics.

Метод емуляції системних бібліотек (library emulation) [232, с. 51; 252] часто асоціюють передусім з API-емуляцією (API emulator), зокрема з

програмним засобом WINE. WINE дозволяє програмному забезпеченню, створеному для ОС Windows, працювати під управлінням Unix-подібних ОС, наприклад, ОС Linux. Це відбувається шляхом перехоплення API-викликів програми та їх емуляції через API-виклики поточної ОС. Розробники WINE наголошують на тому, що це ПЗ не є емулятором (назва «WINE» є рекурсивним акронімом від «WINE is Not an Emulator»), зазначаючи, що WINE є прошарком для забезпечення сумісності між застосунками Windows та POSIX-сумісними ОС, і наголошуючи на принциповій відмінності WINE від повних емуляторів у зв'язку з трансляцією API-викликів безпосередньо у процесі їх виконання без імітації внутрішньої логіки ОС Windows [293]. Водночас, в офіційній документації WINE не запропоновано альтернативного терміну для таких технологій. Термін «емюлятор системних бібліотек», використаний зокрема у роботах [252; 232], дає змогу, з одного боку, показати відмінність описаного вище підходу від технології повної емуляції, а з іншого – об'єднати під спільною назвою WINE та інші наявні й потенційно можливі технології, подібні до нього, наприклад, програмний засіб Cygwin.

Cygwin дає змогу програмам, написаним для ОС Linux, працювати в ОС Windows. При цьому, на противагу WINE, у Cygwin відбувається перекомпіляція Linux-програм із джерельних кодів, а не емуляція API-викликів [208].

Квазіемуляція (quasi-emulation) об'єднує віртуалізацію за методом динамічної трансляції та віртуалізацію за методом паравіртуалізації [280].

Метод *динамічної трансляції* (dynamic translation) також називають методом *бінарної трансляції* (binary translation) та методом *повної віртуалізації* (full virtualization). Для пояснення суті методу необхідно з'ясувати поняття службової інструкції.

Службові інструкції (службові команди, англ. sensitive instructions, дослівно – чутлива команда, чутлива інструкція) – команди ЦП, необхідні для роботи ОС і розраховані на виконання лише на найвищому рівні привілеїв. Якщо службова інструкція надходить від програми, яка працює на нижчому рівні привілеїв, то таку інструкцію може бути виконано інакше або не виконано

взагалі. Прикладами службових інструкцій є команди введення-виведення, команди зміни налаштувань блоку керування пам'яттю (MMU) та ін.

Службові інструкції можуть перехоплюватися на апаратному рівні, як це було свого часу в ОС IBM System 370, однак у хронологічно пізнішій архітектурі x86, розрахованій на ПК, відповідний функціонал не було передбачено. Це стало на заваді створенню ефективних віртуальних машин, коли ідея віртуалізації знову почала набувати популярності. Одним зі шляхів вирішення проблеми службових інструкцій і важливим кроком в еволюції технологій віртуалізації є проект Disco (Стенфордський університет), учасники якого стали засновниками компанії VMware. Відродження віртуальних машин на початку 2000-х років детальніше розглянуто у додатку А.

Метод динамічної трансляції, вперше реалізований у продуктах VMware, ґрунтується на перехопленні службових інструкцій та їх заміні безпечними послідовностями коду. Така заміна відбувається без модифікації коду гостьової ОС, а тому метод підходить для віртуалізації найрізноманітніших ОС, незалежно від того, відкритими є їх програмні коди чи ні.

Метод динамічної трансляції є основним, зокрема, у гіпервізорах VMware Workstation, VMware Player, Oracle VirtualBox.

У разі застосування *методу паравіртуалізації* (paravirtualization), навпаки, відбувається заміна проблемних команд безпосередньо у коді гостьової ОС, а отже, програмні коди цієї ОС мають бути відкритими.

Прикладом віртуалізаційного ПЗ, що спирається на метод паравіртуалізації, є гіпервізор Xen.

Апаратну віртуалізацію (hardware-assisted virtualization, hardware based virtualization) для ПК було відроджено у 2005 році. Сучасна апаратна віртуалізація виконується на базі двох основних методів: Intel VT-x та AMD-V. Обидва методи, попри відмінності у реалізації, ґрунтуються на спільній ідеї – створення контейнерів для віртуальних машин на апаратному рівні. Крім того, попри тривалий період відсутності підтримки апаратної віртуалізації у ЦП для ПК, відповідні технології лишалися доступними у продуктах компанії IBM,

використовуваних як великі обчислювальні потужності (System/390, z-серії) [169].

Варто відзначити, що на практиці технології апаратної віртуалізації не завжди переважають технології програмної віртуалізації за продуктивністю [195]. У сучасних засобах віртуалізації нерідко застосовується поєднання кількох напрямів та методів віртуалізації. Так, гіпервізор II типу KVM застосовує технологію динамічної трансляції для віртуалізації ОС із закритим кодом і технологію паравіртуалізації для віртуалізації ОС із відкритим кодом [241]. Гібридний гіпервізор Oracle VM Server для x86 поєднує технологію паравіртуалізації (для ОС із відкритим кодом) із апаратною віртуалізацією (для ОС із закритим кодом) [263]. Програмний засіб Virtuozzo, що початково спирався на технологію віртуальних контейнерів, згодом також додав до свого функціоналу адаптований гіпервізор KVM [289]. Програмний засіб Proxmox VE використовує віртуальні машини на базі KVM (динамічна трансляція і паравіртуалізація) та віртуальні контейнери на основі OpenVZ [271]. Наведений перелік не є вичерпним і продовжує розширюватися з розвитком технологій віртуалізації.

Однією з відносно нових технологій віртуалізації є **вкладена віртуалізація** (nested virtualization). Вкладена віртуалізація передбачає запуск одного віртуалізованого середовища всередині іншого (наприклад, віртуальної машини всередині віртуальної машини) [204]. Наразі відповідний функціонал доступний лише у деяких засобах віртуалізації і часто потребує ЦП з підтримкою апаратної віртуалізації. Вкладену віртуалізацію реалізовано у гіпервізорах KVM, Xen, VMware ESXi, Microsoft Hyper-V, хмарній платформі Oracle Ravello.

Виходячи із запропонованої систематизації технологій віртуалізації, нами було уточнено значення ряду термінів, зокрема наступних.

Під *засобами віртуалізації* розумітимемо апаратні складові та програмне забезпечення, яке у той чи інший спосіб реалізує три головні риси віртуалізації (поділ ресурсів одного фізичного комп'ютера на декілька взаємно

незалежних віртуальних середовищ або об'єднання ресурсів кількох фізичних комп'ютерів в одне віртуальне середовище; оперативність переходу з одного віртуального середовища в інше; приховування реальних фізичних ресурсів та заміна їх абстракціями).

Відповідне програмне забезпечення називається **програмними засобами віртуалізації**, або **віртуалізаційним програмним забезпеченням**, а відповідні апаратні складові (представлені *процесорами з підтримкою технологій віртуалізації*) – **апаратними засобами віртуалізації**.

Вживаючи термін «засоби віртуалізації» без уточнення, програмними чи апаратними є ці засоби, матимемо на увазі передусім програмні засоби віртуалізації, які, можливо, також передбачають залучення апаратних засобів віртуалізації.

Віртуальну машину розглядаємо як результат віртуалізації, але не як засіб віртуалізації. Програмний засіб віртуалізації, призначений для створення віртуальних машин, називається **гіпервізором**.

Під **технологією віртуалізації** пропонуємо розуміти окремо взятий напрям чи метод віртуалізації, а також їх різномітні поєднання.

Хмарні технології та їх зв'язок з технологіями віртуалізації. Згідно з означенням американського Національного інституту стандартів та технологій (National Institute of Standards and Technology, NIST) [250], **хмарні технології** (*хмарні обчислення, cloud computing*) – це модель, що дозволяє повсюдний, зручний мережний доступ на вимогу (*on-demand access*) до області спільно використовуваних налаштовуваних обчислювальних ресурсів (мереж, серверів, сховищ, застосунків, сервісів тощо), які може бути швидко надано та швидко вивільнено з мінімальними управлінськими затратами та мінімальною взаємодією з постачальником сервісу.

NIST наводить також основні характеристики хмарних технологій, моделі хмарних сервісів та моделі їх розгортання.

Основні характеристики хмарних технологій:

- **самообслуговування на вимогу** – споживач може власноруч

визначати та змінювати обчислювальні потужності, такі як серверний час чи обсяг мережного сховища, через автоматичні засоби, без прямої взаємодії з людиною – представником постачальника ресурсів;

- **універсальний мережний доступ** – обчислювальні потужності доступні по мережі через стандартні механізми, що дозволяє використовувати для доступу до цих потужностей різноманітні пристрої, зокрема настільні ПК, ноутбуки, планшети, смартфони тощо;

- **об'єднання ресурсів** – обчислювальні ресурси постачальника об'єднані для обслуговування багатьох споживачів на основі багатокористувацької моделі; користувач не мусить турбуватися про географічне розміщення використовуваних ним ресурсів, хоча у низці випадків може вибирати країну, регіон, центр даних, у яких розташовуватимуться ці ресурси;

- **еластичність** – обчислювальні потужності динамічно виділяються та вивільняються відповідно до запитів користувача, причому у деяких випадках обсяг ресурсів може зростати і зменшуватись автоматично;

- **вимірюваність обсягів використання сервісу** – автоматичний контроль та оптимізація обсягів використання ресурсів, зазвичай з оплатою за принципом *pay-per-use* та з наданням користувачу можливості моніторингу обсягів використаних ним ресурсів.

Виокремлюють три основні *моделі надання хмарних послуг*.

- **SaaS** (*Software as a Service* – програмне забезпечення як сервіс). Споживач використовує програмні засоби постачальника, які працюють у хмарі. Робота з такими програмними засобами зазвичай доступна з різних пристроїв через веб-переглядач чи спеціалізоване ПЗ. Споживач не має контролю над серверами, операційними системами, сховищами та іншими складовими інфраструктури, у якій працює використовуваний ним програмний засіб, і може змінювати лише обмежений набір користувацьких налаштувань цього ПЗ.

Приклади: Gmail, Google Docs, Google Disk, Google Play Music, Dropbox, Pictochart, Tom's Planner та багато інших.

- **PaaS** (*Platform as a Service* – платформа як сервіс). Споживач

розміщує у хмарній інфраструктурі власні програмні засоби (створені власноруч чи на замовлення, придбані, використовувані за вільними ліцензіями тощо). Для розгортання програмних засобів споживачеві надаються мови програмування, бібліотеки, сервіси та інші інструменти. Як і у випадку SaaS, у PaaS споживач не має контролю над інфраструктурою, але може керувати програмними засобами, що розгортаються, а у низці випадків – також налаштовувати середовище для роботи цих програмних засобів.

Приклади: AWS Elastic Beanstalk, Microsoft Azure Cloud Services, Google App Engine, Oracle Cloud PaaS, Open Shift, Force.com та ін.

- **IaaS** (*Infrastructure as a Service* – інфраструктура як сервіс). Споживачеві надаються засоби для обробки та зберігання даних, мережні засоби та інші фундаментальні обчислювальні ресурси, на базі яких споживач може розміщувати довільне ПЗ. Споживач не керує всією хмарною інфраструктурою, але має контроль над операційними системами, сховищами, програмними засобами, а також, можливо, обмежений контроль над окремими мережними компонентами (наприклад, брандмауерами).

Приклади: Amazon EC2, Google Compute Engine, GoGrid, Microsoft Azure Virtual Machines та ін., а також так зване хмарне ПЗ (OpenStack, OpenNebula та ін.)

Хмарна інфраструктура (не лише у моделі IaaS) може розгортатися відповідно до *чотирьох моделей розгортання*:

- публічна хмара (*public cloud*);
- приватна хмара (*private cloud*);
- спільна хмара (громадська хмара, *community cloud*);
- гібридна хмара (*hybrid cloud*).

Публічну хмару може використовувати будь-хто (але не обов'язково безкоштовно), приватну хмару – лише одна особа або організація, спільну хмару – члени різних організацій, об'єднані спільними цілями. Гібридну хмару складають дві та більше хмар різних моделей.

Хмарні сервіси можуть бути як безкоштовними, так і платними. Поширеним є варіант, коли базовий тарифний план доступний безкоштовно, а

розширені можливості передбачають додаткову плату, причому часто базового функціоналу цілком достатньо для більшості користувачів. Розповсюдженим є варіант, коли сервіс в цілому платний, однак надається безкоштовний пробний період – найчастіше короткий, наприклад, місяць, іноді довший, наприклад, рік (Amazon AWS, Google Cloud Platform, Microsoft Azure).

Залежно від моделі надання послуг та особливостей реалізації, хмарні сервіси у своїй основі використовують розглянуті раніше технології віртуалізації – від віртуалізації програмних застосунків (SaaS) до віртуалізації серверів та цілої інфраструктури на їх основі (IaaS).

Сфери застосування технологій віртуалізації. Вище вже йшлося про віртуалізацію серверів як про один із найпоширеніших напрямів віртуалізації. Е. Таненбаум [168] обґрунтовує доцільність віртуалізації серверів за допомогою наступного прикладу. Нехай маємо мультикомп'ютер, до складу якого входять сервер електронної пошти, веб-сервер, FTP-сервер, кілька серверів електронної торгівлі тощо. Сервери мультикомп'ютера працюють на різних, але об'єднаних у високошвидкісну мережу комп'ютерах. У цій ситуації може справді існувати нагальна потреба у виділенні під кожен сервер окремої машини, зокрема через фізичну неможливість однієї машини впоратися з усіма цими задачами. Проте часто кілька окремих машин використовуються з міркувань безпеки. ОС, у межах якої функціонує той чи інший сервіс, час від часу дає збої. Зрозуміло, що вихід з ладу одного сервісу через збій ОС (кожен сервіс на окремому комп'ютері) є менш критичною ситуацією, ніж вихід з ладу одразу всіх сервісів через збій операційної системи (всі сервіси на одному комп'ютері). Але виділення для роботи кожного сервісу окремого комп'ютера означає також зростання витрат на обладнання (закупівля, встановлення, обслуговування, апгрейдинг, заміна), на живлення, охолодження, вентиляцію, проблеми з розміщенням великої кількості комп'ютерів.

Однак ведучи мову про запуск усіх сервісів на одному комп'ютері, ми дотепер мали на увазі також запуск цих сервісів як процесів під управлінням єдиної ОС. Натомість застосувавши віртуалізацію, одержимо можливість

запустити сервіси на одному комп'ютері, але кожний на окремій віртуальній машині. При цьому ОС на віртуальних машинах можуть співпадати або відрізнятися, але в обох випадках це будуть окремі, незалежні одна від одної операційні системи. Такий підхід знижує ризик одночасного виходу з ладу всіх сервісів, запущених на одному комп'ютері, через збій операційної системи. Зауважимо, що віртуалізована система, як і мультикомп'ютер, підтримує модель часткової відмови, але зі значно меншими затратами і з простішим управлінням [168, с. 661]. Зрозуміло, що лишається загроза одночасного краху всіх сервісів через збій фізичного комп'ютера, на якому ці сервіси запущені, однак, зазначається у [168, с. 661], сервіси частіше виходять з ладу не через відмову обладнання, а через ненадійність програмного забезпечення, у тому числі ненадійність ОС, натомість віртуалізація передбачає, що «єдиною програмою, котра працює в режимі ядра, є гіпервізор, в якого на два порядки менше рядків програмного коду, ніж у повноцінній ОС, а тому і на два порядки менше можливих помилок» [168, с. 661].

Крім того, застосування віртуалізації спрощує міграцію ПЗ з одного фізичного комп'ютера на інший, роблячи можливим переміщення програмного застосунку разом з віртуальною машиною, а отже разом із ОС, бібліотеками та конфігураційними файлами. Переміщення віртуальних машин також дозволяє збалансовувати навантаження між декількома фізичними серверами, причому простіше, ніж за допомогою переміщення процесів, запущених під управлінням невіртуалізованої ОС [3].

Віртуалізація серверів є важливим, проте не єдиним напрямом віртуалізації. Наведемо найпоширеніші напрями та способи застосування технологій віртуалізації на основі виділених О. К. Гультяєвим задач, для розв'язання яких можуть бути використані віртуальні машини [54].

Освоєння нової ОС чи нового ПЗ. Перш ніж встановлювати нову ОС чи нову програму на свій комп'ютер, користувач може попередньо випробувати її на віртуальній машині.

Встановлення кількох ОС на один комп'ютер без використання мультизавантаження. Така можливість забезпечується передусім гіпервізорами II типу: основна ОС лишається незмінною, натомість різних гостьових ОС може бути декілька. У деяких випадках аналогічний ефект забезпечується й гіпервізорами інших типів (Microsoft Hyper-V).

Запуск програмних застосунків, скомпільованих для роботи в ОС, відмінній від поточної. Здійснити такий запуск можна через віртуальні машини (гіпервізор будь-якого типу), а також за допомогою віртуалізації програмних застосунків. Останній підхід фактично робить програму кросплатформеною.

Тестування програмного застосунку, сайту тощо в різних ОС. Проте, зазначається у примітках до видання [168], для гарантування максимальної коректності перевірки рекомендується проводити її завершальний етап на фактичній цільовій платформі, а не на її віртуалізованому варіанті [168, с. 662].

Моделювання обчислювальної мережі на єдиному комп'ютері. Таке моделювання може здійснюватися, зокрема, з навчальною або дослідницькою метою.

На основі аналізу робіт [54; 168; 252] виокремимо ***сфери застосування технологій віртуалізації:***

- системне адміністрування (віртуалізація серверів, віртуалізація робочого столу);
- робота з прикладними програмами (віртуалізація настільних ОС, віртуалізація програмних застосунків);
- розробка та тестування багатоплатформених програмних продуктів (віртуалізація настільних ОС, віртуалізація програмних застосунків);
- веб-дизайн (віртуалізація настільних ОС, віртуалізація програмних застосунків);
- основа для хмарних технологій (віртуалізація серверів, віртуалізація програмних застосунків, віртуалізація робочого столу);
- навчання інформатичних дисциплін, у тому числі самонавчання (віртуалізація настільних ОС, віртуалізація програмних застосунків).

Особливий інтерес для даного дослідження становлять дві останні сфери застосування – навчання інформатичних дисциплін (безпосередньо) та основа для хмарних технологій (опосередковано, як засіб навчання інформатичних дисциплін).

1.1.3. Тлумачення віртуалізації в освіті

Враховуючи, з одного боку, широке розповсюдження терміну «віртуалізація» у різних галузях знань, зокрема в галузях ІКТ та освіти, з'ясуємо, яким чином віртуалізація у досліджуваному нами розумінні співвідноситься з віртуалізацією, про яку йдеться у педагогічних дослідженнях.

Передусім відзначимо, що дослідження проблем віртуалізації освіти здійснюється як із педагогічно-технологічної точки зору (В. Ю. Биков [12], Т. А. Вакалюк [20], С. В. Воєводін [26], П. Ділленбур (P. Dillenbourg) [221], С. Г. Литвинова [88], І. Д. Малицька [93], О. В. Мерзликін [97], М. В. Попель [133], З. С. Сейдаметової [150], М. П. Шишкіної [187] та ін.), так і з суспільно-філософської точки зору (О. Є. Висоцька [24], Д. В. Галкін [30], В. А. Кавалеров [74] та ін.). Суспільно-філософська точка зору послуговується терміном «віртуалізація» передусім у філософському розумінні (віртуалізація як процес переходу від фактичної реальності до реальності потенційної внаслідок певної діяльності людини [19]). Натомість педагогічно-технологічна точка зору переносить поняття віртуалізації у площину теорії та практики застосування у галузі освіти, оперуючи такими поняттями, як віртуальна педагогіка, віртуальна освіта, віртуальна реальність, віртуальний клас, віртуальна лабораторія, віртуальна мережна спільнота тощо. У межах педагогічно-технологічної точки зору на поняття віртуалізації йдеться зокрема (але не виключно) й про віртуальні машини, віртуальні контейнери, віртуалізаційне програмне забезпечення та інші технології, які досліджуються у цій роботі, які відповідають сформульованому раніше робочому означенню віртуалізації (поділ ресурсів одного фізичного комп'ютера на декілька взаємно незалежних віртуальних середовищ чи об'єднання ресурсів кількох фізичних комп'ютерів в одне віртуальне середовище; оперативність переходу з одного віртуального

середовища в інше; приховування реальних фізичних ресурсів та заміна їх абстракціями), і які також можуть бути узагальнені терміном «віртуалізація у вузькому розумінні».

Виокремимо **два основні аспекти зв'язку між віртуалізацією в галузі ІКТ та віртуалізацією в освіті**. Загалом йтиметься про віртуалізацію в різних її розуміннях й за потреби уточнюватиметься, яке саме розуміння мається на увазі. При цьому найбільшу увагу приділено віртуалізації у вузькому розумінні, відповідно до робочого означення віртуалізації.

Перший аспект. *Віртуалізація як засіб навчання.* Розглядаються системи віртуальної реальності та можливості їх застосування в освіті. Належність деякої системи до систем віртуальної реальності визначається передусім її здатністю забезпечити користувачеві ефект присутності в деякому штучному (уявному) середовищі завдяки впливу на органи чуття [94, с. 135]. Використання засобів і технологій систем віртуальної реальності з метою досягнення навчально-виховних цілей досліджується у межах *віртуальної освіти* [8, с. 87-88], котра, у свою чергу, вивчає створення та функціонування систем віртуальної та відкритої освіти [8, с. 84].

Розгляд віртуалізації як засобу навчання може не передбачати безпосереднього використання віртуалізації у вузькому розумінні (як це є у випадку віртуальних класів, віртуальних лабораторій, віртуальних мережних спільнот тощо), а може й передбачати (віртуальні машини, віртуальні контейнери тощо). Та в усіх цих випадках створюється своєрідна віртуальна реальність – віртуальну (уявну) лабораторію, віртуальний (уявний) клас, віртуальний (уявний) комп'ютер всередині комп'ютера фізичного, – а отже, відповідні засоби може бути асоційовано з категорією засобів та технологій систем віртуальної реальності.

Віртуальна педагогіка є складовою *електронної педагогіки*, котра досліджує створення та впровадження ІКТ навчального призначення [8, с. 84]. У Національній доповіді про стан і перспективи розвитку освіти в Україні (2016) інформатизацію освіти визначено імперативом її розвитку, а як одну із

актуальних проблем науково-методичного забезпечення розитку електронної освіти та електронної педагогіки виокремлено реалізацію електронного навчання [106, с. 163].

У випадку, коли програмні засоби віртуалізації використовуються з освітньою метою, такі засоби можуть розглядатися і як ІКТ навчального призначення. Звідси бачимо, що віртуалізаційне ПЗ також може бути досліджене з позицій електронної педагогіки.

Електронна педагогіка розробляє у тому числі й завдання педагогіки відкритої освіти. За В. Ю. Биковим, відкрита освіта передбачає дотримання вісімнадцяти основних принципів: *принципу свободи вибору учня* (вибір спеціальності, рівня освіти, навчальної та індивідуальної програми навчання тощо); *принципу свободи вибору вчителів* (вибір навчальних закладів для роботи, рівня власного навчального навантаження, форм і методів навчання тощо); *принципу гнучкості навчання* (гнучкість індивідуальних планів, методів та форм навчання і т.п.); *принципу інваріантності навчання* (наявність у різних навчальних закладах ряду уніфікованих засобів навчання, педагогічних технологій, навчально-методичних матеріалів); *принципу незалежності навчання у часі* (здійснення навчання у час зручний для учня на для викладача, збільшення часу, відведеного на самостійну роботу); *принципу екстериторіальності навчання* (можливість навчання у момент територіальної віддаленості учня або викладача від навчального закладу); *принципу еквівалентності сертифікатів про освіту* (відповідність національних стандартів міжнародним, а якості відкритої освіти – якості освіти традиційної); *принципу стартового рівня знань* (ідеться зокрема про знання, уміння та навички у галузі ІКТ); *принципу гуманізації навчання* (створення людиноцентристських педагогічних систем, компенсація нестачі «живого» спілкування між учнем і вчителем); *принципу інтернаціоналізації навчання* (виховання поваги до інших культур, усвідомленої потреби опанування іноземних мов, надання можливості участі в міжнародних заходах); *принципу пріоритетності педагогічного підходу* (першочерговість розробки педагогічної

складової під час побудови систем відкритої освіти); *принципу досконалості будови навчального середовища* (еволюційно-поетапне вдосконалення навчального середовища відкритої освіти з метою його максимальної відповідності завданням відкритих педагогічних систем); *принципу економічної привабливості відкритої освіти* (відповідність вартості відкритої освіти фінансовим можливостям потенційних учнів); *принципу несуперечності відкритої освіти* (узгодженість відкритої освіти з уже наявними формами освіти, збереження надбань традиційної освіти); *принципу легітимності відкритої освіти* (створення відповідної законодавчо-правової та методично-інструктивної бази); *принципу престижності відкритої освіти*; *принципу маркетингу освітніх послуг*; *принципу системності створення і розвитку відкритої освіти* [10, с. 48-55]. Відкрита освіта реалізується у відкритих навчальних середовищах.

Засоби, методи та технології, що є напрацюваннями електронної педагогіки (зокрема й віртуалізаційне ПЗ як засіб навчання), можуть бути застосовані для реалізації завдань відкритої освіти, у тому числі стати складовою відкритих навчальних середовищ. Проте зауважимо, що віртуалізаційне ПЗ, як і будь-яке інше ПЗ, є придатним для використання у відкритих навчальних середовищах лише за умови дотримання принципів відкритої освіти.

Зазначимо, що терміни «електронна педагогіка», «віртуальна педагогіка» та подібні поки не належать до усталених. Нерідко вони вживаються передусім у зв'язку з дистанційною освітою. Дистанційна освіта і справді належить до найактивніших споживачів здобутків електронної та віртуальної педагогіки, однак ці здобутки можуть бути використані і в інших формах освіти, зокрема у традиційній [221, с. 9].

Таким чином, програмні засоби віртуалізації можуть бути предметом розгляду електронної та віртуальної педагогіки, а також виконувати роль комп'ютерно орієнтованого засобу навчання в межах системи відкритої освіти.

Другий аспект. Віртуалізація як предмет навчання. Віртуалізація у найрізноманітніших її розуміннях може також становити частину змісту освіти. Так, віртуалізація в ІКТ тлумаченні (у широкому і вузькому розумінні) нерідко розглядається в курсі з ОС у межах лекційної частини курсу і (або) на лабораторних роботах. Якщо при цьому під час виконання студентами завдань лабораторних робіт застосовуються засоби віртуалізації, – маємо поєднання віртуалізації як засобу навчання (перший аспект) та віртуалізації як предмету навчання (другий аспект).

Взаємовідношення між віртуалізацією як засобом навчання й віртуалізацією як предметом навчання проілюстроване на рис. 1.3.



Рис. 1.3. Взаємовідношення між віртуалізацією як засобом навчання і віртуалізацією як предметом навчання

На рис. 1.3 маємо зображення двох множин – множини, елементи якої пов'язані із розглядом віртуалізації як предмету навчання (зовнішній овал), та множини, елементи якої пов'язані із розглядом віртуалізації як засобу навчання

(внутрішній овал). Зовнішній овал повністю містить внутрішній, тобто множини перебувають у відношенні включення. Множина «віртуалізація як засіб навчання» є підмножиною множини «віртуалізація як предмет навчання». На практиці це означає, що коли певна технологія використовується як засіб навчання, то ця технологія водночас може (але не обов'язково мусить) виступати у ролі предмету навчання. Наприклад, застосування віртуальних машин як засобу навчання під час лабораторних робіт – і вивчення технологій, що лежать в основі віртуальних машин (предмет навчання); використання віртуальних класів на уроці (засіб навчання) – вивчення віртуальних класів як технології чи особливостей використання віртуальних класів для роботи з учнями (предмет навчання).

Множину «засіб навчання» на рис. 1.3 візуалізовано як нечітку, тобто наведена діаграма Венна не дає однозначної відповіді на питання про те, чи завжди віртуалізація, що виступає у ролі предмету навчання, водночас може бути використана також і як засіб навчання. Враховуючи широку міждисциплінарність цього питання, воно не було предметом детального розгляду у межах даного дослідження і, безумовно, потребує додаткового вивчення. Втім, у багатьох випадках технології віртуалізації справді належать до обох множин одночасно, і контрприкладу нами не виявлено.

Підсумовуючи сказане вище, сформулюємо наступні висновки про зв'язок між віртуалізацією та освітою.

1) Віртуалізація у різних її розуміннях, зокрема в досліджуваному нами, може становити частину змісту освіти (наприклад, тема «Віртуалізація» у курсі з ОС).

2) Використання засобів віртуалізації як засобів навчання (наприклад, у курсі з ОС) означає також формування відповідних компетентностей.

1.2. Педагогічні особливості навчання операційних систем бакалаврів інформатики

1.2.1. Деякі тенденції сучасної інформатичної освіти

У п. 1.1.3 було розглянуто такі напрями в сучасній освіті, як віртуальна освіта, відкрита освіта, електронна освіта. Усі ці напрями є актуальними у випадку інформатичної освіти. Далі розглянемо фундаменталізацію вищої інформатичної освіти, освіти впровадж життя, компетентнісний, діяльнісний, особистісно орієнтований підходи.

Фундаменталізація вищої освіти. Якщо для освіти 90-х років минулого століття характерним було прагнення підготувати вузькоспеціалізованого фахівця, причому якомога швидше, і це знайшло втілення у диференціації та спеціалізації навчання, то нині пріоритети змінилися. Стрімкі темпи науково-технічного прогресу призводять до скорого застарівання знань, що унеможливорює одержання освіти «раз і назавжди». Аби лишатися конкурентноспроможним на ринку праці, фахівець мусить навчатися постійно, тобто впродовж життя. Крім того, на початку підготовки фахівця неможливо гарантовано передбачити, якими будуть технології на момент завершення його підготовки. Інформаційні технології належать до тих галузей, у котрих описана тенденція особливо відчутна. Така ситуація зумовила тенденції до фундаменталізації вищої освіти, у тому числі і значною мірою – в інформатичній освіті.

Дослідженням питань фундаменталізації інформатичної освіти займалися М. І. Жалдак [63], Т. П. Кобильник [77], В. В. Лаптев [86], М. П. Лапчик [87], Н. В. Морзе [104], Н. І. Рижова [145], С. О. Семеріков [151], М. В. Швецький [86] та ін.

Фундаменталізація передбачає надання майбутньому фахівцю фундаментальних базових знань. При цьому одержані знання повинні бути частиною єдиної світоглядної наукової системи, що відповідає сучасним уявленням про науку та науковим методам [151, с. 22]. Під фундаментальними знаннями розуміють найбільш стабільні та універсальні загальнотеоретичні

знання [86], що дозволяють пізнавати глибинні, сутнісні зв'язки між процесами [151, с. 22]. С. О. Семеріков одним із найперспективніших напрямків фундаменталізації інформатичних дисциплін вважає стабілізацію ядра цих дисциплін, зокрема шляхом орієнтування передусім на стандарти, наприклад, POSIX [151, с. 71-72].

Освіта впродовж життя. Фундаменталізація вищої освіти тісно пов'язана з освітою впродовж життя (навчанням протягом життя). Згадана вище неможливість одержання освіти один раз і на все життя для фахівця означає потребу в періодичній перепідготовці, освоєнні нових технологій, а також необхідність постійного відслідковування новин у галузі та самонавчання.

Документ Європейської Комісії «Перетворимо європейський простір навчання впродовж життя на реальність» (листопад 2001 р.) наводить наступне означення навчання впродовж життя: «Вся діяльність з навчання, здійснювана протягом життя, з метою вдосконалення знань, умінь і компетентностей в інтересах особистісного, громадянського і соціального розвитку, а також / або для отримання роботи» [210].

Питання освіти впродовж життя висвітлене у роботах таких авторів, як В. Ю. Биков [7; 13], А. В. Добридень [59], Б. В. Клименко [162], Л. М. Корчагіна [80], Є. А. Пінчук [129], І. М. Ромашко [13], М. Ф. Степко [162], Л. Л. Товажнянський [162] та ін.

Зазначимо, що коли йдеться про навчання інформаційних технологій, освіті впродовж життя великою мірою сприяє віртуалізація у різних її тлумаченнях, зокрема у розумінні, досліджуваному у даній роботі. Так, завдяки технології віртуальних машин, віртуальних контейнерів, застосованих у вигляді локально встановленого ПЗ чи доступних у хмарі, студенти одержують можливість освоювати нові ОС, ПЗ для цих ОС, відповідні мережні технології не лише у період навчання відповідних дисциплін, а й згодом, у межах самоосвіти, вже маючи досвід практичного використання таких технологій.

Вища інформатична освіта регламентується галузевими стандартами вищої освіти, котрі, в свою чергу, розроблено на основі компетентісного підходу.

Реалізацію **компетентісного підходу** у процесі підготовки вчителів інформатики розглянуто у працях Г. Л. Абдулгалімова [1], М. І. Жалдака [66], О. В. Жмуд [2; 67], Н. В. Морзе [104], К. П. Осадчої [121], Т. В. Отрошко [122; 123], Ю. С. Рамського [66], М. В. Рафальської [66], О. М. Спіріна [158], Ю. В. Триуса [174] та ін.

За О. І. Пометун, компетентність – це «складна інтегрована характеристика особистості, під якою розуміється сукупність знань, умінь, навичок, ставлень, а також досвіду, що разом дає змогу ефективно провадити діяльність або виконувати певні функції, забезпечуючи розв’язання проблем і досягнення певних стандартів у галузі професії або виді діяльності» [132, с. 18].

Згідно із Законом України «Про вищу освіту» [22], компетентність – це «динамічна комбінація знань, вмінь і практичних навичок, способів мислення, професійних, світоглядних і громадянських якостей, морально-етичних цінностей, яка визначає здатність особи успішно здійснювати професійну та подальшу навчальну діяльність і є результатом навчання на певному рівні вищої освіти» [22, ст. 1, п. 13], а освітньо-професійні програми повинні, зокрема, визначати компетентності і результати навчання, якими має оволодіти здобувач [22, ст. 1, п. 17].

Компетентності, які формуються у процесі навчання ОС бакалаврів інформатики, розглянуто у п. 1.2.

Компетентісний підхід пов'язаний із **діяльнісним підходом** (Л. С. Виготський, Д. Б. Ельконін, О. М. Леонтьєв, С. Л. Рубінштейн та ін.) і передбачає спрямованість навчання на формування способу дій, можливе лише у процесі діяльності. Згідно з діяльнісним підходом, навчання відбувається не шляхом передавання знань, а через управління навчальною діяльністю. Формування способів дій досягається шляхом розв'язування навчальних завдань [3].

Особистісно орієнтований підхід у підготовці вчителів інформатики досліджували А. А. Братанич [16], Н. С. Павлова [126], Т. В. Тихонова [171; 172], О. Ю. Усата [176] та ін.

Особистісно орієнтована освіта спрямована на «підтримку й розвиток людини в людині», на становлення учня, студента як самобутньої особистості [139, с. 42]. Така освіта має на меті створити умови, сприятливі для того, аби засвоєння змісту освіти супроводжувалось розвитком особистісних функцій індивіда. Це потребує врахування індивідуально-психологічних особливостей студентів, заохочення їхньої самостійності у навчанні [139, с. 47-48].

Організація навчання ОС із застосуванням засобів віртуалізації відповідно до особистісно орієнтованого підходу передбачає ретельність добору програмних засобів віртуалізації й розробку відповідного навчально-методичного забезпечення так, аби це сприяло самостійному й усвідомленому використанню студентами засобів віртуалізації, забезпечувало підґрунтя для подальшого залучення таких засобів у майбутньому.

1.2.2. Особливості навчання операційних систем у вищій школі

Комп'ютери стрімко входять практично в усі сфери людської діяльності. Це величезне розмаїття пристроїв, від персональних комп'ютерів, ноутбуків, нетбуків, планшетних комп'ютерів, мобільних пристроїв, активно використовуваних дедалі більшою кількістю населення, до мейнфреймів, серверів меншої потужності, а також комп'ютерів, що керують конвеєрами та літальними апаратами, пральними машинами та електронними картками тощо. При цьому навіть найменший комп'ютер керується операційною системою відповідного класу. Отже, ОС становлять так само невід'ємну частину життя сучасної людини, як і комп'ютери, а тому надзвичайно важливим є оволодіння ОС в обсязі, відповідному сфері діяльності. Коли ж ідеться про студентів інформатичних спеціальностей, вимоги розширюються від користування ОС до знання принципів будови та функціонування ОС, здатності виконувати налагодження цих систем, вміння створювати програмне забезпечення для цих систем тощо.

Вагомий сегмент ринку ОС формують Unix-подібні ОС, котрі використовуються на комп'ютерах різних розмірів та призначення. Термін «Unix-подібні ОС» вживають на позначення цілого сімейства ОС, що базуються на ОС Unix та похідних від неї. Це, зокрема, Unix Ware, FreeBSD, Solaris, Linux, OpenBSD, NetBSD, iOS, Android, Minix, Mac OS X та інші системи. Маючи більш ніж сорокарічну історію, Unix-подібні ОС наочно ілюструють як еволюцію розвитку ОС, так і сучасні принципи їх побудови. Так, мову C було створено саме в процесі розробки UNIX. Крім того, серед Unix-подібних ОС є велика кількість безкоштовних, що виключає додаткові витрати на придбання системи. У зв'язку з описаними вище обставинами, Unix-подібні ОС нерідко є частиною змісту низки інформатичних дисциплін у вищій школі.

Аналіз спеціальностей (раніше – напрямів підготовки) та дисциплін, у межах яких здійснюється вивчення ОС (у тому числі Unix-подібних) протягом даного дослідження здійснювалося двічі:

- 1) на підготовчому етапі дослідження (2010-2012 рр.) і на початку констатувального етапу дослідження (2013-2015 рр.);
- 2) на формуальному етапі (2015-2018 рр.).

Потребу повторного аналізу було зумовлено змінами нормативно-правової бази щодо вищої освіти, зокрема прийняттям Постанови Кабінетів Міністрів України «Про затвердження переліку галузей знань і спеціальностей, за якими здійснюється підготовка здобувачів вищої освіти» [73].

У процесі аналізу вітчизняного досвіду навчання ОС, проведеного протягом *підготовчого* (2010-2012 рр.) та *констатувального* (2013-2015 рр.) етапів дослідження, з'ясовано, що вивчення ОС здійснювалося за напрямками підготовки 6.040301 «Прикладна математика», 6.040302 «Інформатика», 6.040303 «Системний аналіз» (галузь знань 0403 «Системні науки та кібернетика»); 6.050101 «Комп'ютерні науки», 6.050102 «Комп'ютерна інженерія», 6.050103 «Програмна інженерія» (галузь знань 0501 «Інформатика та обчислювальна техніка»); 6.170101 «Безпека інформаційних і комунікаційних систем», 6.170102 «Системи технічного захисту інформації»,

6.170103 «Управління інформаційною безпекою» (галузь знань 1701 «Інформаційна безпека»), 6.050201 «Системна інженерія», 6.050202 «Автоматизація та комп'ютерно-інтегровані технології» (галузь знань 0502 «Автоматика та управління») та ін. Вивчення ОС, у тому числі Unix-подібних ОС у проаналізованих випадках відбувалося у межах дисциплін та спецкурсів «Операційні системи», «Операційні системи та системне програмування», «Системне програмування та операційні системи», «Системне програмування», «Системне програмне забезпечення», «Системне адміністрування ОС Linux», «Інформаційні системи» тощо.

Під час повторного аналізу на *формуальному* етапі дослідження (2015-2018 рр.) виявлено, що вивчення ОС передбачене спеціальностями 014.09 «Середня освіта (Інформатика)» (галузь знань 01 «Освіта/Педагогіка»); 113 «Прикладна математика» (галузь знань 11 «Математика та статистика»); 121 «Інженерія програмного забезпечення», 122 «Комп'ютерні науки», 123 «Комп'ютерна інженерія», 125 «Кібербезпека», 126 «Інформаційні системи та технології» (галузь знань 12 «Інформаційні технології»); 151 «Автоматизація та комп'ютерно-інтегровані технології» (галузь знань 15 «Автоматизація та приладобудування») та ін. у межах дисциплін та спецкурсів «Операційні системи та системне програмування», «Операційні системи», «Системне адміністрування Linux», «Системне програмне забезпечення», «Системне програмування», «Системне програмування та операційні системи», «Сучасні мережеві операційні системи», «Системне програмування та системне програмне забезпечення», «Адміністрування і налагодження комп'ютерних систем», «Мережні операційні системи», «Системне адміністрування», «Unix-подібні операційні системи», «Розробка додатків та адміністрування в ОС Linux», «Основи операційних систем» та ін.

Зауважимо однак, що перелічені вище курси не обов'язково передбачають розгляд саме Unix-подібних ОС. У деяких випадках у навчальній програмі курсу зазначається лише вивчення ОС Windows. І навпаки, курс може бути цілком присвячений Unix-подібним ОС. Надалі йтиметься про курси, що

поєднують вивчення Unix-подібних ОС з вивченням інших ОС (наприклад, Windows), або курси, котрі присвячені лише Unix-подібним ОС.

Фактичні дані порівняння курсів з ОС у ЗВО України станом на 2018 рік наведено у додатку В. У ході дослідження з'ясовувалися відповіді на наступні питання.

- Для студентів яких спеціальностей викладається курс з ОС?
- Чи здійснюється у ЗВО підготовка за напрямом 6.040302 «Інформатика» / спеціальністю 014.09 «Інформатика»?
- Як називається навчальна дисципліна (дисципліни), у межах якої (яких) здійснюється навчання ОС?
- Чи використовуються засоби віртуалізації під час проведення лабораторних робіт, і якщо так, то які саме?

Загалом досліджено відомості про курс з ОС у 22 ЗВО України, під час добору яких перевага надавалася ЗВО – лідерам за вітчизняними рейтингами, зокрема за підготовкою ІТ-фахівців (рейтинги DOU 2018, «ТОП-200 Україна», рейтинг за даними наукометричної бази даних SciVerse Scopus 2018 року), а також ЗВО, які здійснюють підготовку за спеціальністю 014.09 «Середня освіта (Інформатика)», і / або ЗВО, щодо яких було знайдено інформацію про застосування цими закладами засобів віртуалізації у межах курсу з ОС. Таким чином, розглянуто відомості про курс з ОС у Вінницькому національному технічному університеті, Дніпропетровському національному університеті імені Олеся Гончара, Донецькому національному університеті імені Василя Стуса (м. Вінниця), Київському національному університеті імені Тараса Шевченка, Криворізькому державному педагогічному університеті, Львівському національному університеті імені Івана Франка, Національному авіаційному університеті, Національному аерокосмічному університеті імені М. Є. Жуковського («ХАІ»), Національному педагогічному університеті імені М. П. Драгоманова, Національному технічному університеті «Харківський політехнічний інститут», Національному технічному університеті «Дніпровська політехніка», Національному університеті «Львівська політехніка»,

Національному університеті «Києво-Могилянська академія», Національному університеті біоресурсів і природокористування України, НТУУ «Київський політехнічний інститут імені Ігоря Сікорського» («КПІ»), Одеському національному політехнічному університеті, Сумському державному університеті, Тернопільському національному педагогічному університеті імені Володимира Гнатюка, Харківському національному економічному університеті імені Семена Кузнеця, Харківському національному університеті радіоелектроніки, Чернівецькому національному університеті імені Юрія Федьковича, Чорноморському національному університеті імені Петра Могили (додаток В). Використано інформацію, одержану з офіційних веб-ресурсів ЗВО, наукових, науково-методичних та навчально-методичних публікацій, а також під час усних та електронних бесід з викладачами та випускниками ЗВО.

З'ясовано перелік основних спеціальностей, якими передбачене проходження курсу з ОС, а також варіанти назви відповідних навчальних дисциплін (див. вище).

Серед засобів віртуалізації, застосовуваних під час лабораторних робіт з ОС, виявлено як засоби, встановлювані локально (гіпервізор VirtualBox), так і надання студентам віддаленого доступу до віртуальних машин та віртуальних контейнерів на базі віртуалізованого сервера й у хмарі (додаток В). Водночас зазначимо, що у значній частині випадків на офіційних ресурсах ЗВО та в доступних для вільного завантаження навчально-методичних матеріалах була відсутня інформація про використовувані засоби віртуалізації. У деяких випадках вказано, що такі засоби застосовуються, однак не конкретизовано, які саме. Втім, на основі анкетування викладачів курсу з ОС у ЗВО України (див. п. 4.2) маємо підстави констатувати розповсюдженість використання засобів використання для навчання ОС та різноманітність цих засобів.

Згідно з навчальними планами, що діяли протягом підготовчого та констатувального етапів дослідження для напряму підготовки 6.040302 «Інформатика» Житомирського державного університету імені Івана Франка, дисципліну «Операційні системи та системне програмування»

визначено як вибірккову дисципліну з циклу професійної та практичної підготовки. Курсом передбачено надання базових теоретичних відомостей щодо внутрішньої організації сучасних ОС, підготовка студентів до адміністрування цих систем та створення ПЗ для них. Одержані протягом вивчення дисципліни знання та навички мають практичне застосування у роботі вчителя (викладача) інформатики, а також є початковою базою для діяльності системного адміністратора, прикладного програміста, системного програміста тощо. Курс ґрунтується на знаннях, отриманих під час вивчення дисциплін «Програмування», «Комп'ютерні мережі», «Алгоритми і структури даних», «Математична логіка і теорія алгоритмів». Станом на вказаний період навчальним планом дисципліни було передбачено: вивчення дисципліни на лекційних та лабораторних заняттях, самостійна робота студентів; перевірка основних теоретичних знань та практичних умінь студентів за допомогою аудиторних контрольних робіт; складання екзамену. Оскільки Житомирський державний університет імені Івана Франка входив до експериментальної бази даного дослідження (див. розділ 4), у процесі дослідження навчальну та робочу програму дисципліни «Операційні системи та системне програмування» було змінено відповідно до розробленої методики застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики. Навчальну та робочу програму дисципліни, застосовувані протягом формувального етапу педагогічного експерименту (2015-16 н. р.), наведено у додатку Г.

Дослідженню процесу підготовки бакалаврів інформатики присвячені роботи Т. А. Вакалюк [20], Т. Я. Вдовичин [21], Н. Б. Єпик, А. П. Кузьменко, В. М. Кузьменко [83], У. П. Когут [78] та ін. Підготовку майбутніх вчителів інформатики досліджували В. Ю. Биков [13], А. М. Гуржій [55], Л. В. Брескіна [18], М. І. Жалдак [64], М. П. Лапчик [87], Н. В. Морзе [251], С. М. Прийма [136], Ю. С. Рамський [142], О. М. Спирін [158; 159], Т. В. Тихонова [171], Ю. В. Триус [174], Г. Ю. Цибко [182], Г. В. Шугайло [188] та ін.

Навчання операційних систем у закладає вищої освіти присвячено наукові

та навчально-методичні публікації А. Є. Батюка [5], М. Ф. Бондаренка [15], Д. Є. Ванькевича [5], В. М. Волохова [27], В. Ю. Габрусєва [29; 85], Я. М. Глинського [33], В. В. Дьомкіна [61], Г. Г. Злобіна [5], О. Г. Качко [15], В. В. Лапінського [85; 29], Т. В. Панченка [127], Х. В. Паньків [128], Н. А. Рибачок [143], В. А. Рязської [33], А. М. Стрюка [164], В. А. Шеховцова [185] та ін.

Загальну структуру та орієнтовну класифікацію компетентностей вчителя інформатики було обґрунтовано О. М. Спіріним [158, с.212]. Визначено загальні компетентності (компетентності індивідуальної ідентифікації й саморозвитку, міжособистісні компетентності, суспільно-системні компетентності) та професійно-спеціалізовані компетентності (загальнопрофесійні, предметно-орієнтовані, технологічні, професійно-практичні). Серед компонентів системи професійно-спеціалізованих компетентностей також виокремлено *інформатичні компетентності* [157].

О. М. Спірін визначає **інформатичну компетентність** як «підтверджену здатність особистості задовольнити власні індивідуальні потреби і суспільні вимоги щодо формування професійно-спеціалізованих компетентностей людини в галузі інформатики» [157].

У запропонованій у [157] системі професійно-спеціалізованих компетентностей інформатичні компетентності, пов'язані з операційними системами, належать до предметно-орієнтованих (а саме науково-предметних) та до професійно-практичних компетентностей. Предметно-орієнтована компетентність стосується профільної професійної діяльності майбутнього випускника (у випадку вчителів – діяльності з навчання певного предмета) [158, с. 213]. Професійно-практична компетенція відповідає конкретним практичним роботам, які вимагаються від випускника з боку роботодавця [158, с. 214].

У контексті курсу з ОС здійснюється формування передусім наступних науково-предметних та професійно-практичних компетентностей.

Науково-предметні компетентності [158, с. 221]:

- вміти логічно і послідовно подати засвоєні знання теоретичних основ інформатики й історії її розвитку;

- демонструвати знання основ інформаційних систем (архітектура, апаратне і програмне забезпечення ЕОМ, локальні та глобальні комп'ютерні мережі, мультимедіа тощо);
- вміти розв'язувати типові прикладні математичні, статистичні й соціально-економічні задачі, ефективно використовувати сучасне системне (операційні системи; системи процедурного, функціонального, логічного, об'єктивно-орієнтованого програмування, мови Інтернет-публікацій; сервісні програми) і прикладне (загальне, спеціальне) програмне забезпечення ЕОМ;
- мати уявлення про інформаційну безпеку і вміти здійснювати елементарний захист даних на електронних носіях;
- вміти самостійно здійснювати пошук та аналіз необхідних повідомлень у контексті розвитку предметної галузі інформатики.

Професійно-практичні компетентності [158, с. 225]:

- уміти встановлювати, налагоджувати сучасні версії ОС, поширене прикладне та спеціальне програмне забезпечення ПК;
- демонструвати розвинуте програмування щонайменше однією з сучасних об'єктно-орієнтованих мов, наприклад C++.

Враховуючи сказане вище, **інформатичні компетентності щодо операційних систем (ОС)** пропонуємо розуміти як підтверджені на практиці здатності особистості на основі опанованих знань, умінь та навичок з операційних систем використовувати такі системи та технології їх адміністрування й системного програмування для задоволення власних індивідуальних потреб і розв'язування професійних задач у галузі інформатики.

Під час підготовчого етапу дослідження (2010-2012 рр.) та на початку констатувального етапу дослідження (2013-2015 рр.) діяв галузевий стандарт вищої освіти України для напряму підготовки 6.040302 «Інформатика», основними складовими якого були освітньо-кваліфікаційна характеристика випускника вищого навчального закладу (ОКХ) [98] та освітньо-професійна програма (ОПП) [99]. Постановою Кабінету Міністрів України «Про

затвердження переліку галузей знань і спеціальностей, за якими здійснюється підготовка здобувачів вищої освіти» [73] затверджено новий перелік галузей знань та спеціальностей, зокрема спеціальність 014.09 «Середня освіта (Інформатика)» галузі знань 01 «Освіта/Педагогіка», що прийшла на зміну напряму підготовки *Інформатика*, і нові вимоги до стандартів вищої освіти. Водночас, на момент завершення дослідження стандарт для спеціальності 014.09 «Середня освіта (Інформатика)» не затверджено.

Враховуючи описані вище обставини, встановлено зв'язок між інформатичними компетентностями щодо операційних систем майбутніх учителів інформатики [158], компетентностями бакалаврів інформатики з галузевого стандарту для напряму підготовки 6.040302 «Інформатика» [98] та компетентностями із проекту галузевого стандарту для спеціальності 014.09 «Середня освіта (Інформатика)» [100], які формуються у межах курсу з ОС (додаток Д).

На основі проаналізованих компетентностей виокремимо наступні інформатичні **компетентності** бакалаврів інформатики щодо ОС:

- знати та розуміти основні етапи, напрями та тенденції розвитку ОС;
- орієнтуватися у класифікації ОС, з урахуванням їх архітектури та сфери застосування;
- знати та розуміти теоретичні основи будови та функціонування ОС;
- вміти використовувати інтерфейс користувача Unix-подібних ОС (графічний інтерфейс, інтерфейс командного рядка);
- вміти здійснювати налаштування мережного з'єднання у Unix-подібних ОС;
- вміти встановлювати програмне забезпечення у Unix-подібних ОС, виконувати його наступне оновлення та деінсталяцію;
- вміти відстежувати роботу процесів та потоків в ОС і керувати ними;
- вміти здійснювати моніторинг використання різних видів пам'яті в ОС;
- володіти основами роботи з файловою системою у Unix-подібних ОС;
- володіти основами роботи з налаштуваннями системи безпеки ОС;

- володіти основами системного програмування мовою C++;
- вміти одержувати довідкові відомості щодо використання ОС та програмування у них.

1.2.3. Педагогічні умови навчання операційних систем

У психології та педагогіці, окрім навчання, розглядають також научіння – діяльність, спрямовану на отримання знань. У такому разі навчання постає як доцільно організоване научіння. Виокремлюють асоціативну, умовно-рефлекторну, знакову та операційну концепції научіння [72]. У процесі навчання ОС спиратимемося передусім на операційну концепцію научіння. Крім того, за потреби звертатимемося також до асоціативного принципу.

Операційна концепція (П. Я. Гальперін, Н. Ф. Тализіна) та тісно пов'язана з нею *теорія поетапного формування розумових дій* розглядають діяльність як основу пізнавальних процесів. Діяльність, поетапно еволюціонуючи від предметних до розумових дій, у результаті стає абстрактним пізнанням [72, с. 94]. У зв'язку зі сказаним вище, великого значення під час навчання ОС набуває *діяльнісний підхід* (Л. С. Виготський, Д. Б. Ельконін, О. М. Леонт'єв, С. Л. Рубінштейн).

Асоціативний принцип полягає у введенні нових знань, спираючись на вже наявні знання завдяки існуванню певного роду асоціативних зв'язків між старими і новими знаннями. Зауважимо, що асоціативний принцип, хоч і пов'язаний з асоціативною концепцією, однак їй не тотожний. Асоціативна концепція зводить до взаємодії відчуттів та уявлень взагалі весь процес навчання, тимчасом як асоціативний принцип апелює до чуттєвого досвіду суб'єкта пізнання лише в окремі моменти [72, с. 86]. У випадку навчання ОС асоціативний принцип може бути успішно застосований під час введення ряду абстрактних понять (процес, потік, адресний простір, системний виклик тощо), для пояснення принципу роботи операцій (наприклад, перемикання контекстів),

алгоритмів (зокрема, годинникового алгоритму заміщення сторінок) тощо [75; 168].

Педагогічними умовами навчання дисципліни будемо називати обставини, які мають регулярний вплив на перебіг процесу навчання цієї дисципліни.

Серед педагогічних умов навчання ОС визначимо зовнішні та внутрішні умови. Зовнішні умови визначаються професіоналізмом діяльності викладача, організацією навчального процесу у ЗВО, навчальними програмами, навчальними планами та відповідними галузевими стандартами вищої освіти. Внутрішні умови полягають у розвитку особистісних рис студентів та в його здатності до організації власної пізнавальної діяльності. Обидві категорії умов перебувають у взаємозв'язку: зовнішні умови покликані сприяти розвитку внутрішніх, внутрішні – підвищувати ефективність зовнішніх.

Внутрішніми умовами навчання ОС у підготовці бакалаврів інформатики вважатимемо: сформованість мотиваційної сфери студента; розвиток логічного мислення студента; графічну та алгоритмічну культуру студента; рівень сформованості компетентностей студента з інших інформатичних дисциплін (у тому числі з архітектури ЕОМ, програмування, теорії алгоритмів та математичної логіки), з іноземних мов (передусім англійської), а також з фізико-математичних дисциплін; пізнавальні вміння студента (аналітичні, проектувальні, конструктивні); емоційно-позитивне ставлення студента до учіння; здатність студента до самоорганізації; зорієнтованість щодо майбутньої професійної діяльності; спроможність до співпраці на рівноправних засадах; здатність студента до підпорядкування зі зворотним зв'язком.

До *зовнішніх умов* навчання ОС віднесемо професіоналізм діяльності викладача; мікросоціум навчальної групи; мікросоціум закладу освіти; навчальні приміщення освітнього закладу; галузеві стандарти вищої освіти; цілепокладання; побудову навчальних планів, програм та змісту навчального матеріалу; форми, методи і засоби навчання).

Окреслені вище зовнішні умови навчання ОС відповідають навчальному середовищу та чинникам глобального освітнього простору, описаним В. Ю. Биковим [9; 11]. Навчальне середовище утворюють учнівсько-групова складова, вчительська складова та система засобів навчання. До глобального освітнього простору В. Ю. Биков відносить складову навчального закладу (мікросоціум навчального закладу, а також навчальні приміщення), освітньо-просторову складову (люди та суспільні системи за межами навчального закладу, як причетні до системи освіти, так і не причетні до неї, котрі, однак, впливають чи можуть впливати на навчальну діяльність).

Визначені умови навчання ОС бакалаврів інформатики було враховано під час розробки методики застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики. Неоднорідність певної частини цих умов (як внутрішніх, так і зовнішніх, див. п. 3.3, 4.2) стала однією з причин вироблення варіативного підходу до застосування таких технологій (п. 3.3).

1.3. Аналіз закордонного досвіду навчання операційних систем у вищій школі

Закордонний досвід навчання ОС у ЗВО досліджувався шляхом опрацювання науково-методичних публікацій, підручників, навчальних програм, рекомендацій щодо розробки навчальних програм, а також відомостей з сайтів закладів освіти.

Викладанню курсу з ОС присвячено роботи таких закордонних авторів, як Ч. Андерсон (C. L. Anderson) [197], Дж. Андрус (J. Andrus) [198], Р. Арчер (R. Archer) [298], А. Авів (A. J. Aviv) [200], Д. Брилов (D. Brylow) [205], Г. Бушенеб (H. Boucheneb) [225], К. Вейл (C. Vaill) [254], Н. Вієно (N. Viennot) [242], Р. Гесс (R. Hess) [230], А. Гармпіс (A. Garmpis) [224], Ф. Гіральдо (F. Giraldeau) [225], Т. Големанов (T. Golemanov) [226], Е. Големанова (E. Golemanova) [226], Н. Гувацос (N. Gouvatsos) [224], К. Граммер (K. Grammer) [227], М. Даренаіс (M. R. Dagenais) [225], К. Далл (C. Dall) [220], Д. Джонс (D. Jones) [236], С. Дюгнан (S. Duignan) [222], О. Лаадан (O. Laadan)

[242], Я. Лі (Y. Li) [298], Ц. Лю (J. Liu) [247], Б. Лу (B. T. Loo) [200], В. Манніно (V. Mannino) [200], Ф. Мачадо (F. B. Machado) [248], Л. Мая (L. P. Maia) [248], М. Нгуєн (M. Nguyen) [197], Дж. Ні (J. Nieh) [198; 220; 242; 253; 254], Е. Ньюман (A. Newman) [236], Т. Оуларн (T. Owlarn) [200], Дж. Оуєнс (J. Owens) [266], А. Палічерла (A. Palicherla) [267], Е. Пачеко (A. C. Pacheco Jr.) [248], Б. Піоро (B. Pioro) [298], П. Полсон (P. Paulson) [230], Д. Портер (D. Porter) [267], Дж. Столерман (J. Stolerman) [227], К. Сюй (K. Xu) [200], Т. Чжан (T. Zhang) [267], Л. Шамір (L. Shamir) [274], С. Шеннін (S. Shannin) [200], С. Юань (X. Yuan) [298], Б. Юї (B. Yi) [227] та ін.

Закордонну практику використання технологій віртуалізації у навчанні ОС представлено у роботах авторів Г. Бушенеб (H. Boucheneb) [225], М. Даренаїс (M. R. Dagenais) [225], С. Дюгнан (S. Duignan) [222], А. Гармпіс (A. Garmpis) [224], Ф. Гіральдо (F. Giraldeau) [225], Н. Гуавацос (N. Gouvatsos) [224], О. Лаадан (O. Laadan) [242], Дж. Ні (J. Nieh) [198; 220; 242; 253; 254], К. Вейл (C. Vaill) [254], Н. Вієно (N. Viennot) [242] та ін. У статті [254] розглянуто досвід викладання вступного курсу з ОС із застосуванням технологій віртуальних машин та віддаленого робочого столу задля організації роботи студентів над проектами з розробки ядра ОС, у роботі [242] описано підхід до проведення курсу з ОС на основі готових образів віртуальних машин на базі VMware Workstation/Fusion та системи розподіленого контролю версій Git. У статті [225] висвітлено підхід до введення базових понять курсу з ОС для студентів спеціальності «Computer Engineering» (комп'ютерна інженерія), який передбачає, зокрема, застосування віртуальних машин для програмування драйверів. Роботу [222] присвячено використанню віртуальних машин у навчанні ОС студентів спеціальності «Business Computing» (інформатика у бізнесі). У статті [224] пропонується веб-ресурс, який забезпечує доступ до емульованої ОС Ubuntu Linux.

Важливо відзначити, що розгляд закордонного досвіду навчання ОС для аналізу ускладнювався відсутністю у закордонній вищій освіті точного аналога предметної спеціалізації 014.09 «Середня освіта (Інформатика)» [73], якому у

період аналізу закордонного досвіду відповідав напрям підготовки 6.040302 «Інформатика».

Спільною цільовою групою представників ACM (Association for Computing Machinery – Асоціація обчислювальної техніки), AIS (Association for Information Systems – Асоціація інформаційних систем) та комп'ютерного товариства IEEE (IEEE Computer Society) виокремлено п'ять комп'ютерних спеціальностей: «Computer Science» (інформатика), «Computer Engineering» (комп'ютерна інженерія), «Information Systems» (інформаційні системи), «Information Technologies» (інформаційні технології), «Software Engineering» (програмна інженерія) [211]. Ближчою до перелічених спеціальностей є вітчизняна галузь знань 12 «Інформаційні технології». Втім, попри відмінність професійно-практичної складової підготовки за предметною спеціалізацією 014.09 «Середня освіта (Інформатика)» та за спеціальностями галузі знань 12 «Інформаційні технології», існує значна подібність переліку дисциплін, які вивчаються. Тож вважаємо за доцільне розглянути досвід навчання ОС у межах спеціальностей, які можуть бути узагальнені спільним терміном «computing», у межах даного дослідження.

З п'яти висвітлених у CS2005 спеціальностей найближчою до спеціальності 014.09 «Середня освіта (Інформатика)» є спеціальність «Computer Science», що надає «всеосяжну основу, яка дозволяє випускникам адаптуватися до нових технологій та ідей» [211, с. 14]. Водночас про таку подібність є сенс говорити лише з урахуванням відмінностей щодо професійно-практичної складової, оскільки предметна спеціалізація 014.09 «Середня освіта (Інформатика)» передбачає застосування згаданої основи передусім у галузі освіти, а отже, розглядає технічну складову менш глибоко, ніж це передбачено інженерними спеціальностями.

Рекомендації щодо змісту та результатів освітньої діяльності за спеціальністю «Computer Science» викладено у керівних принципах розробки навчальних програм Computer Science Curricula 2013 (далі – CS2013) [213], які є результатом роботи спільної цільової групи представників ACM та IEEE. У

CS2013 неодноразово підкреслюється зорієнтованість цього документу на гнучкість, а отже, можливість його використання у різних закладах освіти та для різних спеціальностей.

У додатку Е наведено фрагмент CS2013, присвячений галузі знань «Операційні системи» [213, с. 135-141], перекладений нами з англійської. Фрагмент містить загальний опис галузі знань, змістові модулі та питання, рекомендовані до вивчення, очікувані результати навчання тощо.

В основі CS2013 лежать наступні принципи [213, с. 20-22]:

- розвиток у студентах здатності доповнювати свої знання знаннями з інших дисциплін;
- підготовка випускників до різноманітних професій, включаючи міждисциплінарні галузі;
- надання рекомендацій щодо очікуваного рівня опанування тем;
- забезпечення реалістичних рекомендацій, які справді можуть бути враховані на практиці і також дозволяють включати нові актуальні теми;
- можливість застосування у широкому колі закладів освіти;
- регулювання обсягу основного матеріалу з метою недопущення збільшення загального обсягу навчального навантаження;
- підготовка студентів до навчання впродовж життя;
- визначення базових знань, умінь і навичок і, водночас, забезпечення якнайвищої гнучкості у виборі тем;
- забезпечення максимальної гнучкості щодо об'єднання тем у курси і навчальні плани;
- розробка і перегляд CS2013 повинні передбачати участь широкого кола представників цільової аудиторії та враховувати наступний відзив від них.

Також документ визначає такі очікувані характеристики випускників [213, с. 23-26]:

- технічне розуміння інформатики;
- ознайомленість з загальними темами та принципами інформатики;

- розуміння взаємозв'язку між теорією та практикою;
- системність мислення;
- навички вирішення проблем;
- досвід роботи над проектами;
- готовність до навчання впродовж життя;
- готовність нести професійну відповідальність;
- комунікаційні та організаційні навички, включаючи навички самоорганізації;
- усвідомлення широти сфери застосування комп'ютерів;
- розуміння залежності знань від конкретних галузей та здатність до професійного спілкування з експертами з інших галузей.

Змістові модулі, пропонувані для вивчення у CS2013, структуровано за галузями знань (Knowledge Areas, KAs). Назви галузей знань можуть співпадати з назвами дисциплін, однак автори документу наголошують, що така відповідність не обов'язково мусить бути повною. Коло питань, висвітлених в окремому курсі, може як загалом збігатися з переліком змістових модулів, визначених у певній галузі знань, так і бути ширшим чи вужчим за цю галузь. Один курс може торкатися тем з кількох галузей знань, а одна галузь знань – поєднувати матеріал, що вивчається у межах різних курсів. У CS2013 свідомо не надається рекомендацій щодо розподілу навчального матеріалу за курсами, це питання лишається на розсуд авторів навчальних планів та навчальних програм. Натомість, у додатках до CS2013 наводяться приклади програм реальних курсів, які відповідають визначеним у документі вимогам, і автори яких виявили готовність поділитися своїми напрацюваннями із широким загалом.

Описаний вище підхід стосується і навчання ОС. Зокрема, у CS2013 зазначається, що «курс з операційних систем може бути спроектовано так, щоб охоплювати більш загальні системні принципи, такі як програмування мовами низького рівня, паралелізм та синхронізація, вимірювання продуктивності, комп'ютерна безпека, поряд із темами, котрі стосуються власне операційних

систем» [213, с. 28-29]. Тобто містити теми не лише з однойменної галузі знань (OS: Operating Systems), а й з низки суміжних.

Рекомендовані змістові модулі поділено на нормативні (Core) та вибіркові (Elective). Нормативні змістові модулі, у свою чергу, можуть бути визначені як такі, що належать до нормативних змістових модулів першого рівня (Core Tier-1) чи до нормативних змістових модулів другого рівня (Core Tier-2). Автори CS2013 вважають за необхідне включення до навчальних програм:

- всіх нормативних змістових модулів з Core Tier-1 (цей матеріал мають опанувати всі студенти);
- всіх чи майже всіх змістових модулів з Core Tier-2 (всі студенти повинні освоїти більшість цього матеріалу);
- значну частину вибірових змістових модулів (автори наполягають на недостатності включення лише нормативних змістових модулів).

В описах обсягів годин, рекомендованих для виділення на вивчення окремих змістових модулів, CS2013 орієнтований на лекційні години, лишаючи розподіл решти годин на розсуд авторів навчальних планів та навчальних програм. При цьому у документі окремо зазначено, що такий підхід не передбачає неодмінного застосування виключно традиційних лекцій [213, с. 33].

У CS2013 навчальні досягнення студентів описуються трьома рівнями [213, с. 34].

1) **Ознайомленість (Familiarity).** Студент розуміє суть поняття. Даний рівень характеризується базовою ознайомленістю з поняттям на противагу здатності продуктивно використовувати його на практиці. Рівень передбачає знання відповіді на питання «Що ви про це знаєте?»

2) **Застосування (Usage).** Студент здатний використовувати поняття у певний спосіб (правильно застосовувати деяку синтаксичну конструкцію у програмах, певний метод – у доведеннях тощо). Рівень передбачає знання відповіді на питання «Що ви знаєте про те, як це можна зробити?»

3) **Володіння (Assessment).** Студент здатний розглядати поняття з різних точок зору і/або обґрунтовувати вибір того чи іншого підходу для розв'язування

задачі. Рівень передбачає знання відповіді на питання «Чому треба зробити саме так?»

Як уже було сказано вище, курс з ОС, згідно з CS2013, може включати висвітлення не лише тем з галузі знань «Операційні системи» (Operating Systems, OS), а й тем з інших галузей знань, зокрема із галузі знань «Основи систем» (System Fundamentals, SA), «Безпека інформації» (Information Assurance and Security, IAS), «Архітектура та організація» (Architecture and Organization, AR), «Мережі та комунікація» (Networking and Communication, NC), «Паралельні та розподілені обчислення» (Parallel and Distributed Computing, PD).

У табл. 1.2 подано перелік нормативних та вибіркового змістових модулів з галузі знань «Операційні системи» із рекомендованим обсягом годин.

Таблиця 1.2

Нормативні та вибірково змістові модулі галузі знань «Операційні системи» (Operating Systems, OS) відповідно до CS2013

№	Назва змістового модуля	Статус змістового модуля: нормативний (Core Tier-1/ Core Tier-2) чи вибірково	Мінімальний обсяг лекційних годин (лише для нормативних змістових модулів), год
1.	Огляд операційних систем	нормативний (Core Tier-1)	2
2.	Основні принципи роботи операційної системи	нормативний (Core Tier-1)	1
3.	Паралелізм	нормативний (Core Tier-2)	1,5
4.	Планування та диспетчеризація	нормативний (Core Tier-2)	2
5.	Керування пам'яттю	нормативний (Core Tier-2)	2
6.	Безпека і захист	нормативний (Core Tier-2)	2
7.	Віртуальні машини	вибірково	—
8.	Керування пристроями	вибірково	—
9.	Файлові системи	вибірково	—
10.	Системи реального часу та вбудовані системи	вибірково	—
11.	Відмовостійкість	вибірково	—
12.	Оцінювання продуктивності системи	вибірково	—
Загальний обсяг лекційних годин (без урахування вибіркового змістових модулів):			10,5

Раніше у даній роботі було виокремлено два основні аспекти зв'язку між віртуалізацією в інформатиці й віртуалізацією в освіті: віртуалізація як засіб навчання (забезпечення ефекту віртуальної реальності у тому чи іншому вигляді) та віртуалізація як предмет навчання (вивчення віртуалізації у межах низки дисциплін). Технології віртуалізації досліджуються у цій роботі передусім з точки зору першого аспекту (засіб навчання), однак водночас важливим є розуміння студентами суті застосовуваних ними технологій. Тож доцільно розглянути і другий аспект (віртуалізація як предмет навчання), а отже, й ознайомитися з відповідним закордонним досвідом.

З табл. 1.2 видно, що у CS2013 змістовий модуль «Віртуальні машини» рекомендується як вибірковий і, як вказано далі у документі [213, с. 139], передбачає розгляд таких питань, як: типи віртуалізації; віртуальна пам'ять і сторінкова організація пам'яті; віртуальні файлові системи; гіпервізори; портативна віртуалізація; емуляція та ізоляція; вартість віртуалізації. Тобто у межах змістового модуля може вивчатися не лише віртуалізація у досліджуваному нами розумінні, а віртуалізація у ширшому розумінні (віртуальна пам'ять, віртуальні файлові системи тощо). Водночас, використання технологій віртуалізації як засобів навчання даним документом не регламентується.

Під час аналізу досвіду викладання курсу з ОС у ЗВО світу брався до уваги передусім досвід ЗВО, які лідирують у рейтингах за підготовкою зі спеціальності «Computer Science». Окрім того, до розгляду було включено досвід низки закладів освіти, рейтинги яких дещо нижчі, проте які публікували відомості про свій курс з ОС у той чи інший спосіб. Також досліджено досвід проведення курсу з ОС на масових відкритих онлайн-курсах. Аналіз здійснювався на основі матеріалів курсу, розміщених на веб-ресурсах відповідних університетів; навчальних програм, поданих у додатках до CS2013; науково-методичних публікацій, згаданих раніше у цьому підрозділі.

Було розглянуто курси з ОС, що викладаються в Аерокосмічному університеті Ембрі-Ріддла (Embry-Riddle Aeronautical University), США; Гарвардському університеті (Harvard University), США; Гельсінському

університеті (University of Helsinki), Фінляндія; Єльському університеті (Yale University), США; Кембриджському університеті (University of Cambridge), Великобританія; Коледжі Вільямса (Williams College), США; Колумбійському університеті (Columbia University), США; Массачусетському технологічному інституті (Massachusetts Institute of Technology, MIT), США; Принстонському університеті (Princeton University), США; Рейк'явіцькому університеті (Reykjavik University), Ісландія; Стенфордському університеті (Stanford University), США; Технологічному інституті Джорджії (Georgia Institute of Technology, Georgia Tech), США; Університеті Арканзасу, Літл Рок (University of Arkansas at Little Rock), США; Університеті Каліфорнії, Берклі (University of California, Berkley), США; Університеті Карнегі-Меллона (Carnegie-Mellon University, CMU), США; Федеральній вищій школі Цюриха (Swiss Federal Institute of Technology, ETH Zurich), Швейцарія, а також курс з ОС на базі системи масових відкритих онлайн-курсів Udacity.

З'ясовувалося, які теми вивчаються; які ОС розглядаються; як організовано практичну частину курсу; яке місце у навчанні ОС займають технології віртуалізації; які засоби віртуалізації використовуються.

Фактичні дані порівняльного аналізу курсів з ОС у ЗВО світу наведено у додатку Ж. Рис. 1.4 демонструє розподіл курсів з ОС у проаналізованих ЗВО світу за застосовуваними засобами віртуалізації. Символом * позначено заклади освіти, в яких вказаний засіб/засоби віртуалізації використовуються як допоміжні. Єльський університет вказано у двох стовпчиках і без символу *, оскільки QEMU і mCertiKOS застосовуються разом (mCertiKOS поєднує риси гіпервізора й ОС). На рис. 1.5 подано розподіл курсів ОС у проаналізованих ЗВО за ОС, вивчення яких передбачене курсом.

Серед виявлених особливостей розглянутих курсів з ОС зазначимо такі.

Розгляд технологій віртуалізації у межах лекційної частини курсу.

Програмами значної частини розглянутих курсів з ОС передбачено вивчення віртуалізації під час лекцій. При цьому можливий як розгляд технологій віртуалізації у досліджуваному нами розумінні, так і у ширшому розумінні.

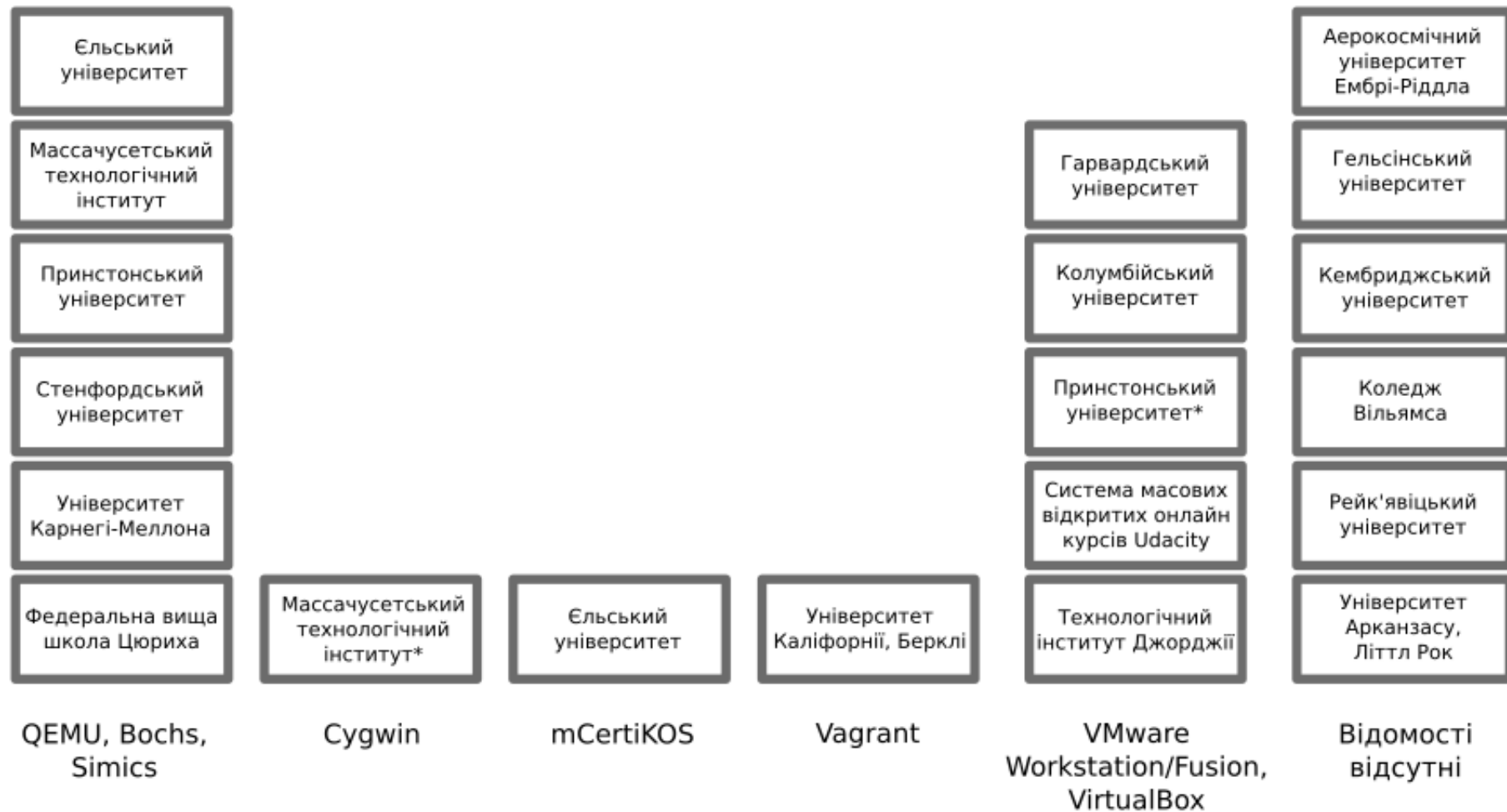


Рис. 1.4. Розподіл курсів з ОС у проаналізованих ЗВО світу за застосовуваними засобами віртуалізації

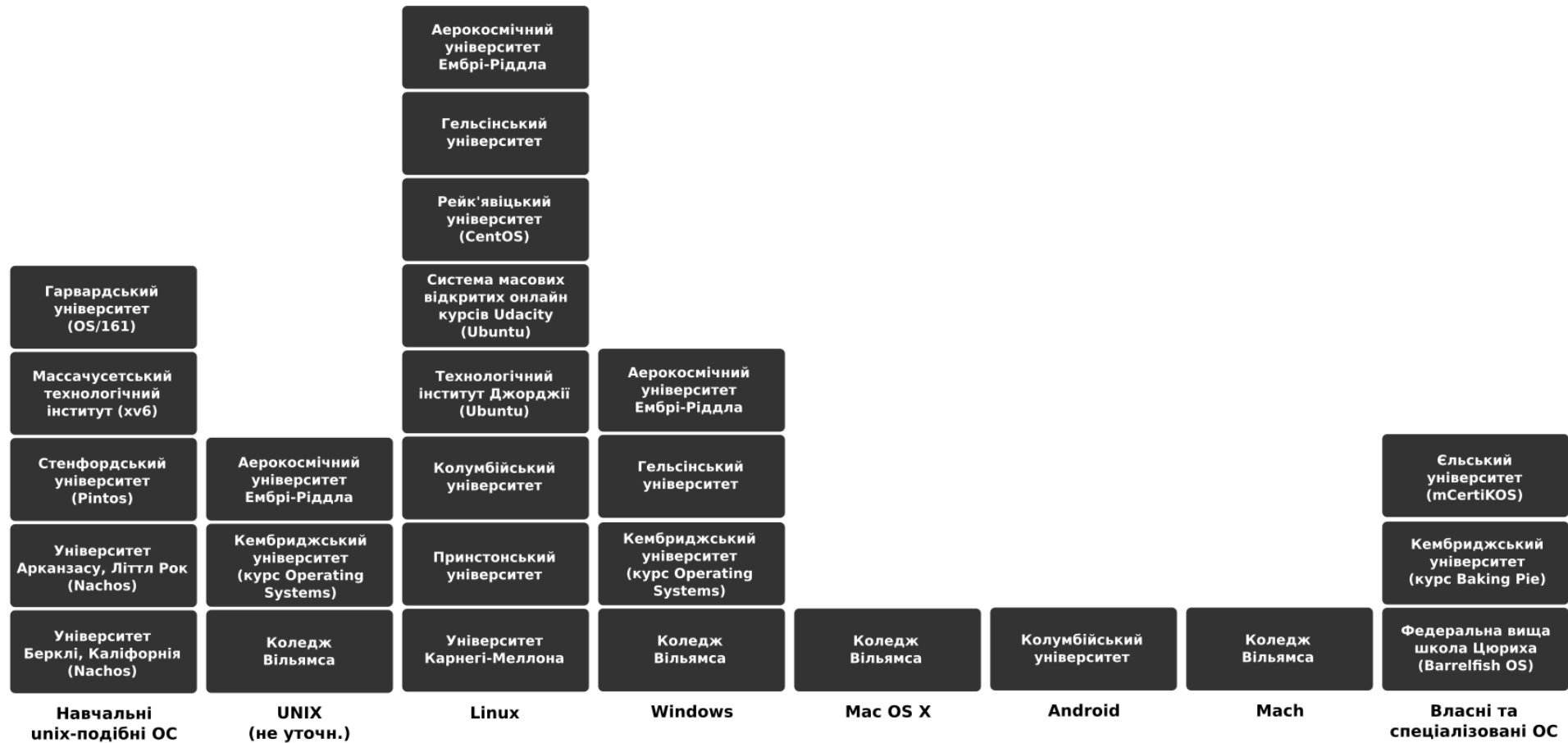


Рис. 1.5. Розподіл курсів з ОС у проаналізованих ЗВО світу за ОС, які вивчаються

Застосування технологій віртуалізації під час виконання лабораторних робіт. З рис. 1.4 видно, що досліджені курси з ОС передбачають застосування досить різноманітних технологій віртуалізації. Використовуються повні емулятори (QEMU, Bochs, Simics), гіпервізори II типу (VMware Workstation, VMware Fusion, VirtualBox), віртуальне середовище розробки Vagrant, спеціалізований гіпервізор для безпечних хмарних технологій mCertiKOS і, додатково, емулятор системних бібліотек Cygwin.

Для частини курсів нами не знайдено відомостей про використання технологій віртуалізації, що може бути зумовлене як обмеженістю пошуків виключно відкритими джерелами, так і тим, що засоби віртуалізації справді не застосовуються (використовується спеціально відведена комп'ютерна лабораторія; курсом передбачене створення простої ОС для одноплатного комп'ютера Raspberry Pi тощо).

Зазначимо також, що не підтверджується існування і наявності на перший погляд тенденції використання закладами освіти – лідерами рейтингів з підготовки за спеціальністю «Computer Science» – виключно повних емуляторів. Такі засоби справді застосовуються як основні у Массачусетському технологічному інституті, Стенфордському університеті, університеті Карнегі-Меллона, Принстонському та Єльському університетах, Федеральній вищій школі Цюриха. Водночас, в інших закладах освіти, зокрема у Гарвардському та Колумбійському університетах, основними використовуваними засобами віртуалізації є гіпервізори II типу, а курс, що викладається в університеті Каліфорнії, Берклі, передбачає роботу з віртуальним середовищем розробки Vagrant.

Пріоритетність вивчення Unix-подібних ОС. Абсолютна більшість розглянутих курсів передбачає вивчення Unix-подібних ОС (рис. 1.5). Водночас конкретні Unix-подібні ОС, що вивчаються, різняться і можуть бути навчальними Unix-подібними ОС (Pintos, Nachos, OS/161, xv6), дистрибутивами Linux (Ubuntu, CentOS тощо), Mac OS X (у ролі додаткової ОС, поряд з іншою, не пропрієтарною Unix-подібною ОС), мобільними Unix-подібними ОС

(Android). Зазначимо також, що у межах деяких курсів паралельно відбувається вивчення й ОС інших платформ (зокрема Windows та Mach).

Поєднання у практичній частині курсу системного програмування для ОС та (у деяких випадках) їх адміністрування. У багатьох проаналізованих випадках практичну частину курсу з ОС зорієнтовано на написання/зміну компонентів ОС (найчастіше мовою C, проте також мовою C++ і Assembler) чи навіть написання власної простої операційної системи для комп'ютера Raspberry Pie («Baking Pi – Operating Systems Development», Кембриджський університет). У деяких випадках у курсі спостерігався помітний акцент на паралельні та розподілені обчислення («Operating Systems», Кембриджський університет). Втім, CS2013 не обмежує галузь знань «Operating Systems» теорією ОС та практикою їх створення. До цієї галузі знань автори документу також відносять, скажімо, такі змістові модулі, як «ОС/Безпека та захист»(OS/Security and Protection, Core-Tier2), що передбачає, у тому числі, вивчення механізмів політики та методів безпеки ОС, питань аутентифікації, розмежування доступу, захисту, резервного копіювання, роботи з обліковими записами тощо.

Передбаченість курсом лабораторних робіт. Майже в усіх випадках курс ОС має практичне спрямування і передбачає виконання студентами лабораторних робіт (проектів з програмування, практичних завдань, домашніх завдань, уроків тощо). Винятком серед розглянутих курсів є теоретичний курс «Операційні системи» (Operating Systems) Кембриджського університету, що виступає у ролі основи для інших, більш поглиблених курсів, які, однак, уже передбачають наявність у студента бакалаврського ступеня, а тому виходять за межі даного дослідження.

Навчання основ роботи із застосовуванням ПЗ, зокрема з програмними засобами віртуалізації. Відповідні інструктивні та інструктивно-методичні матеріали (розроблені спеціально для курсу або універсальні) розміщені у вільному онлайн-доступі і розраховані на самостійне опрацювання студентами. Майже в усіх інших випадках у тексті лабораторної

роботи за потреби наводяться посилання на описи основ роботи з тим чи іншим програмним засобом, часто – написані спеціально для студентів цього навчального закладу або цього курсу.

Доступність альтернативних засобів віртуалізації. Веб-ресурси кількох курсів містять рекомендації щодо засобів віртуалізації, відмінних від основних засобів, використовуваних у межах курсу. Ці рекомендації адресовані студентам, які мають комп'ютери альтернативних апаратних та операційних платформ або які є власниками комп'ютерів з обсягом системних ресурсів, недостатнім для нормальної роботи основних засобів віртуалізації, застосовуваних під час цього курсу. Частина курсів пропонує різні комбінації програмних засобів, у тому числі програмних засобів віртуалізації, залежно від того, якою ОС користується студент (окремі засоби для Linux/Windows/Mac OS X), системних характеристик апаратної складової його комп'ютера (окремі засоби для комп'ютерів з 32-розрядними процесорами та 32-розрядними версіями ОС). Крім того, принаймні в одному випадку (Принстонський університет) безпосередньо вказано можливість віддаленого доступу до комп'ютерної лабораторії, у якій проходять заняття з ОС, та надано інструкції для такого доступу.

Чіткі терміни здачі лабораторних робіт. У більшості випадків розклад курсу містить конкретні дати, до яких необхідно здати ту чи іншу роботу.

Заздалегідь визначена політика щодо списування. Окремо наголошується також на необхідності вказувати джерела за умови використання готових фрагментів програмного коду (наприклад, коду із відкритих джерел). Описуються такі ситуації, як співпраця студентів під час групових проектів.

Використання мережних ресурсів для захисту лабораторних робіт, зокрема Git (Колумбійський університет), персональний простір на сервері університету (Університет Карнегі-Меллона), Dropbox (Принстонський університет). Доступ до опублікованих файлів надається лише викладачам, які приймають роботи.

Виявлені шляхи використання технологій віртуалізації було враховано під час побудови методики застосування технологій віртуалізації Unix-подібних ОС у навчанні бакалаврів інформатики.

Висновки до розділу 1

У розділі 1 описано результати теоретичного аналізу стану розробленості проблеми у науково-методичній, психолого-педагогічній, технічній, навчальній літературі, у практиці навчання ОС у ЗВО.

Враховуючи широке розповсюдження та різноманітність тлумачень терміну «віртуалізація», сформульовано робоче означення віртуалізації, згідно з яким *віртуалізація* – це поняття, що об'єднує технології, засоби, методи тощо, котрим притаманні три головні риси: (1) поділ ресурсів одного фізичного комп'ютера на декілька взаємно незалежних віртуальних середовищ або об'єднання ресурсів кількох фізичних комп'ютерів в одне віртуальне середовище; (2) оперативність переходу з одного віртуального середовища в інше; (3) приховування реальних фізичних ресурсів та заміна їх абстракціями. Під *засобами віртуалізації* варто розуміти апаратні складові та програмне забезпечення, які у той чи інший спосіб реалізують три головні риси віртуалізації, перелічені вище. Відповідне програмне забезпечення є *програмними засобами віртуалізації*, а відповідні апаратні складові – *апаратними засобами віртуалізації*. *Технологією віртуалізації* пропонуємо називати окремо взятий напрям чи метод віртуалізації, а також їх різноманітні поєднання.

Розгляд низки підходів до систематизації технологій віртуалізації дав змогу виробити узагальнену систематизацію цих технологій. Систематизація здійснювалася за двома критеріями: напрям віртуалізації та метод віртуалізації. За напрямом віртуалізації виокремлено віртуалізацію серверів, віртуалізацію настільних ОС, віртуалізацію програмних застосунків та віртуалізацію робочого столу. За методом віртуалізації виокремлено програмну та апаратну віртуалізацію. Апаратна віртуалізація виконується на основі двох основних

технологій: Intel VT-x та AMD-V, програмна віртуалізація – методом повної емуляції, емуляції системних бібліотек чи квазіемуляції. Квазіемуляцію може бути реалізовано методом динамічної трансляції та методом паравіртуалізації.

Виокремлено два основні аспекти зв'язку між віртуалізацією в галузі ІКТ та віртуалізацією в освіті: (І) віртуалізація як засіб навчання (технології, котрі у тому чи іншому вигляді забезпечують ефект віртуальної реальності, використані як засіб навчання – віртуальні класи, віртуальні лабораторії, віртуальні мережні спільноти, віртуальні машини, віртуальні контейнери тощо); (ІІ) віртуалізація як предмет навчання (віртуалізація у будь-якому з її тлумачень як частина змісту освіти). Визначено часткову відповідність віртуалізації у досліджуваному розумінні обох аспектам.

Досліджено відомості про курс з ОС у вітчизняних ЗВО (інформація з офіційних веб-ресурсів ЗВО, наукові, науково-методичні та навчально-методичні публікації, усні та електронні бесіди з викладачами та випускниками). З'ясовано перелік основних спеціальностей, студенти яких вивчають курс з ОС, варіанти назви відповідних навчальних дисциплін. Виявлено застосування під час лабораторних робіт з ОС як встановлювані локально засоби віртуалізації, так і надання віддаленого доступу до віртуальних машин і віртуальних контейнерів на базі віртуалізованого сервера та у хмарі.

Інформатичні компетентності бакалаврів інформатики щодо ОС пропонуємо розуміти як підтверджені на практиці здатності особистості на основі опанованих знань, умінь та навичок з операційних систем використовувати такі системи та технології їх адміністрування й системного програмування для задоволення власних індивідуальних потреб і розв'язування професійних задач у галузі інформатики. Виокремлено інформатичні компетентності бакалаврів інформатики щодо ОС: знати та розуміти основні етапи, напрями та тенденції розвитку ОС; орієнтуватися у класифікації ОС, з урахуванням їх архітектури та сфери застосування; знати та розуміти теоретичні основи будови та функціонування ОС; вміти використовувати інтерфейс користувача Unix-подібних ОС (графічний інтерфейс і/або

інтерфейс командного рядка); вміти налаштовувати мережне з'єднання у Unix-подібних ОС; вміти встановлювати програмне забезпечення у Unix-подібних ОС, виконувати його наступне оновлення та деінсталяцію; вміти відстежувати роботу процесів та потоків в ОС і керувати ними; вміти здійснювати моніторинг використання різних видів пам'яті; володіти основами роботи з файловою системою у Unix-подібних ОС; володіти основами роботи з налаштуваннями системи безпеки ОС; володіти основами системного програмування мовою C++ у Linux і/або у Windows; вміти одержувати довідкові відомості щодо використання ОС та програмування у них.

Визначено педагогічні умови навчання ОС бакалаврів інформатики: зовнішні (професіоналізм діяльності викладача; мікросоціум навчальної групи; мікросоціум навчального закладу; навчальні приміщення освітнього закладу; галузеві стандарти вищої освіти; цілепокладання; побудову навчальних планів, програм та змісту навчального матеріалу; форми, методи і засоби навчання) та внутрішні (сформованість мотиваційної сфери студента; розвиток логічного мислення студента; графічну та алгоритмічну культуру студента; рівень сформованості в студента компетентностей з інших інформатичних дисциплін, фізико-математичних дисциплін та з іноземних мов; пізнавальні вміння студента; емоційно-позитивне ставлення студента до учіння; здатність студента до самоорганізації; зорієнтованість щодо майбутньої професійної діяльності; спроможність до співпраці на рівноправних засадах; здатність студента до підпорядкування зі зворотним зв'язком).

Аналіз закордонного досвіду навчання ОС у вищій школі (науково-методичні публікації, підручники, навчальні програми, рекомендації щодо розробки навчальних програм Computer Science Curricula 2013, відомості з онлайн ресурсів навчальних закладів). Серед виявлених особливостей досліджених курсів з ОС: пріоритетність вивчення Unix-подібних ОС; поєднання у практичній частині курсу системного програмування для ОС та (у деяких випадках) їх адміністрування; передбаченість курсом лабораторних робіт; розгляд технологій віртуалізації у межах лекційної частини курсу;

поширеність застосування засобів віртуалізації під час виконання лабораторних робіт та різноманітність цих засобів; навчання основ роботи із застосовуваним ПЗ (зокрема із засобами віртуалізації); доступність альтернативних засобів віртуалізації та ін. Виявлені шляхи використання технологій віртуалізації було враховано під час розробки методики застосування технологій віртуалізації Unix-подібних ОС у навчанні бакалаврів інформатики.

Основні результати здійсненого у розділі 1 дослідження опубліковано у роботах [37; 38; 46; 48; 49; 50].

РОЗДІЛ 2.

ПРОЕКТУВАННЯ МЕТОДИКИ ЗАСТОСУВАННЯ ТЕХНОЛОГІЙ ВІРТУАЛІЗАЦІЇ UNIX-ПОДІБНИХ ОПЕРАЦІЙНИХ СИСТЕМ У ПІДГОТОВЦІ БАКАЛАВРІВ ІНФОРМАТИКИ

2.1. Загальна методика дослідження

В основу дослідження покладено ідею про потребу індивідуального та усвідомленого добору засобів віртуалізації Unix-подібних систем для підготовки бакалаврів інформатики. У ході дослідження зазначену ідею було продовжено й доповнено ідеєю про існування підстав та можливостей для поєднання кількох засобів віртуалізації у межах одного курсу з операційних систем та необхідність відповідного навчально-методичного забезпечення.

Гіпотеза дослідження: якщо навчання Unix-подібних операційних систем здійснювати на основі розробленої методики застосування технологій віртуалізації, то це дасть змогу підвищити результативність формування інформатичних компетентностей щодо операційних систем у бакалаврів інформатики.

Методологічну основу дослідження складають компетентісний підхід в освіті, положення про фундаменталізацію вищої інформатичної освіти, відкриту освіту та освіту впродовж життя, особистісно орієнтований та діяльнісний підхід у навчанні.

Організація дослідження. Дослідження здійснювалося у період з 2010 по 2018 рр. й проходило у наступні етапи.

Підготовчий етап (2010-2012 рр.). Протягом цього етапу виокремлено суперечності, притаманні процесу застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики, сформульовано проблему, об'єкт, предмет, мету і завдання дослідження; здійснено добір основних методів дослідження; проаналізовано науково-методичну, психолого-педагогічну, технічну, навчальну літературу з тематики дослідження; з'ясовано та уточнено основні терміни; побудовано узагальнену систематизацію технологій віртуалізації.

Констатувальний етап (2013-2015 рр.). На даному етапі проаналізовано вітчизняний та закордонний досвід навчання ОС та роль у ньому засобів віртуалізації; визначено педагогічні умови навчання ОС; сформульовано критерії добору програмних засобів віртуалізації у навчанні Unix-подібних ОС; проведено констатувальний етап педагогічного експерименту та проаналізовано його результати; уточнено гіпотезу дослідження; розроблено модель застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики; вироблено варіативний підхід до застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики, відповідно до якого підібрано комбінацію засобів віртуалізації для курсу з ОС для бакалаврів інформатики й розроблено методичний посібник «Операційні системи та системне програмування».

Формувальний етап (2015-2018 рр.). У цей період проведено формувальний етап педагогічного експерименту; здійснено аналіз та узагальнення одержаних експериментальних даних; розроблено методичні рекомендації «Технології віртуалізації у навчанні операційних систем бакалаврів інформатики», що підсумовують результати дослідження; проведено експертне опитування з метою вибору змістових модулів для опису на їх прикладі розробленої методики; сформульовано загальні висновки та окреслено перспективи подальших досліджень; впроваджено результати дослідження у процес підготовки бакалаврів інформатики у вітчизняних ЗВО.

Експериментальною базою дослідження на різних етапах проведення педагогічного експерименту виступали Житомирський державний університет імені Івана Франка, Дрогобицький державний педагогічний університет імені Івана Франка, Сумський державний педагогічний університет імені А. С. Макаренка.

Вірогідність результатів дослідження забезпечується методологічними основами дослідження, відповідністю методів дослідження його меті і завданням, кількісним і якісним аналізом теоретичного та емпіричного матеріалу, результатами педагогічного експерименту.

2.2. Модель застосування технологій віртуалізації Unix-подібних операційних систем у підготовці бакалаврів інформатики

Модель є одним із можливих результатів *педагогічного проектування* [116] як важливого компоненту професійної діяльності викладача.

Проблеми педагогічного проектування у різний час досліджували В. С. Безрукова [6], Д. Джонс [56], Я. Дітріх [57], В. В. Докучаєва [60], О. С. Заір-Бек [68], О. М. Коберник [76], І. І. Коновальчук [79], А. О. Лігоцький [89], З. Ф. Мазур [91], В. М. Монахов [103], Г. Є. Муравйова [105], П. І. Образцов [116], О. Г. Прикот [137], В. Є. Радіонов [140], Т. Ю. Подобєдова [131], О. М. Соломатін [156], Н. М. Суртаєва [166], В. Ю. Стрельніков [163], Н. О. Яковлева [191], В. А. Ясвін [192] та інші.

На думку О. Заір-Бек, педагогічним проектуванням є «творчість як створення й реалізація педагогічних задумів, спрямованих на вдосконалення освіти, її розвиток і перетворення в конкретних умовах» [68, с. 25].

У [113] проектуванням називають «діяльність, під якою мається на увазі у гранично стислій характеристиці промислювання того, що має бути».

За П. І. Образцовим, педагогічне проектування означає розробку дидактичного опису педагогічної системи, реалізація якої передбачається в рамках навчального процесу [116, с. 9].

В. Ю. Стрельніков виділяє два трактування педагогічного проектування [163, с. 21]:

- **педагогічне проектування у широкому розумінні** передбачає конструювання теоретичних та нормативних моделей на базі загальної теорії;
- **педагогічне проектування у вузькому розумінні** означає створення окремих проектів, що безпосередньо скеровують практичну навчальну діяльність.

Враховуючи практичну спрямованість даного дослідження, надалі нас передусім цікавитиме *педагогічне проектування у вузькому розумінні*.

Часто поняття «педагогічне проектування» тісно пов'язане з поняттям «проектування технології», однак має місце і поняття «проектування методичної системи», «проектування методики» та ін. Дещо детальніше зупинимося на деяких із цих понять.

Методика (методична система) складається з п'яти взаємопов'язаних компонентів: цілі, зміст, форми, методи і засоби навчання [154, с. 8], відповідаючи на запитання «навіщо вчити» (мета), «що вчити» (зміст), «як вчити» (форми, методи і засоби) [153, с. 662].

Технологія навчання, за визначенням П. І. Образцового, є науковим проектом дидактичного процесу, «точне відтворення якого гарантує успіх педагогічних дій». Роль результуючого проекту може виконувати опис дидактичного процесу або його *модель* [116]. С. О. Сисоєва відзначає, що технології, у порівнянні з методикою, притаманна відтворюваність та стійкість результатів. У методиці на перший план виходять «цільовий, змістовий, якісний і варіативно-орієнтовний аспекти», тимчасом як у технології – «процесуальний, кількісний і розрахунковий компоненти» [153, с. 662]. П. І. Образцов називає технологію вищою стадією розвитку методики, наголошує на персоніфікованому характері технології та порівнює її з авторською методикою навчання, проте наголошує на можливості реалізації технології не лише самим її автором [116].

Як уже зазначалося, нерідко дослідники наголошують на саме *технологічному* аспекті проектування, маючи на увазі, що результатом проектування є створення певної навчальної технології (В. С. Безрукова [6, с. 95-96], В. В. Докучаєва [60], Г. Є. Муравйова [105, с. 14], В. М. Монахов [103] та ін.). Однак у низці випадків є підстави також говорити і про *проектування методики*. Це передусім стосується випадку, коли йдеться про *часткову, або власну, технологію навчання*, що «описує процес навчання в масштабі окремого знання», результатом якої є «гарантоване набуття особою знань про певний об'єкт навчання, умінь і навичок практично використовувати їх у майбутній професійній діяльності» [155, с. 106] і яка також постає шляхом

попереднього проектування. Описана вище технологія є процесуальною складовою методичної системи [155, с. 106]. Саме такий підхід до проектування відповідає меті й завданням даного дослідження.

Отже, під проектуванням методики застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики розумітимемо процес розробки *проекту методики застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики*.

Головною метою проектування методики застосування технологій віртуалізації Unix-подібних операційних систем у підготовці бакалаврів інформатики є створення *моделі застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики*.

За В. Ю. Биковим, *модель* – це «деяке подання (аналог, образ) системи, що моделюється, в якому відображаються, враховуються, характеризуються і можуть відтворюватися такі особливості й властивості цієї системи, які забезпечують досягнення цілей побудови та використання моделі» [8, с. 232].

У даному дослідженні *об'єктом моделювання* є процес застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики, здійснюваний у межах курсу з ОС. Враховуючи це, надалі розглядатимемо **модель застосування технологій віртуалізації Unix-подібних операційних система у підготовці бакалаврів інформатики** як модель, що описує організацію навчального процесу і містить структуровані цілі, методичні шляхи їх досягнення, очікувані результати навчального процесу та засоби їх діагностування.

В основі пропонованої моделі застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики лежить послідовність проектування відповідної методики. Розіб'ємо дану послідовність на **п'ять етапів**.

- 1) Постановка мети та завдань.
- 2) Формування змісту.
- 3) Розроблення методики.

- 4) Діагностування проблем.
- 5) Аналіз результатів діагностування.

Поставивши у відповідність кожному етапу проектування методики окремий компонент моделі, одержуємо наступні компоненти моделі застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики.

- 1) Цільовий компонент.
- 2) Змістовий компонент.
- 3) Технологічний компонент.
- 4) Діагностичний компонент.
- 5) Аналітичний компонент.

Водночас, становлення навчальної методики є тривалим процесом. Оптимальна побудова курсу досягається шляхом його щорічного перегляду та доопрацювання. Крім того, більшості інформатичних дисциплін, у тому числі курсу з ОС, притаманний стрімкий розвиток, а отже – швидке застарівання й потреба у регулярному оновленні.

Отже, щороку методика підлягає перегляду щонайменше з двох причин:

- 1) через потребу вдосконалення методичних компонентів курсу з урахуванням досвіду попереднього року (років);
- 2) через появу нових досягнень у відповідній галузі.

Якщо курс викладається неодноразово (наприклад, щороку), послідовність етапів 1-5 повторюється, й на кожному етапі можуть вноситися зміни до мети і завдань (етап 1), змісту (етап 2), форм, методів та засобів (етап 3), щоразу відбувається діагностування (етап 4) та аналіз її результатів (етап 5). Загальне схематичне подання описаної послідовності етапів проектування методики наведено на рис. 2.1.

Після виконання останнього етапу (аналіз результатів) здійснюється повернення до першого етапу (мета та завдання), що покладає початок наступного, $(i+1)$ -го повтору циклу, або $(i+1)$ -ї *ітерації*.

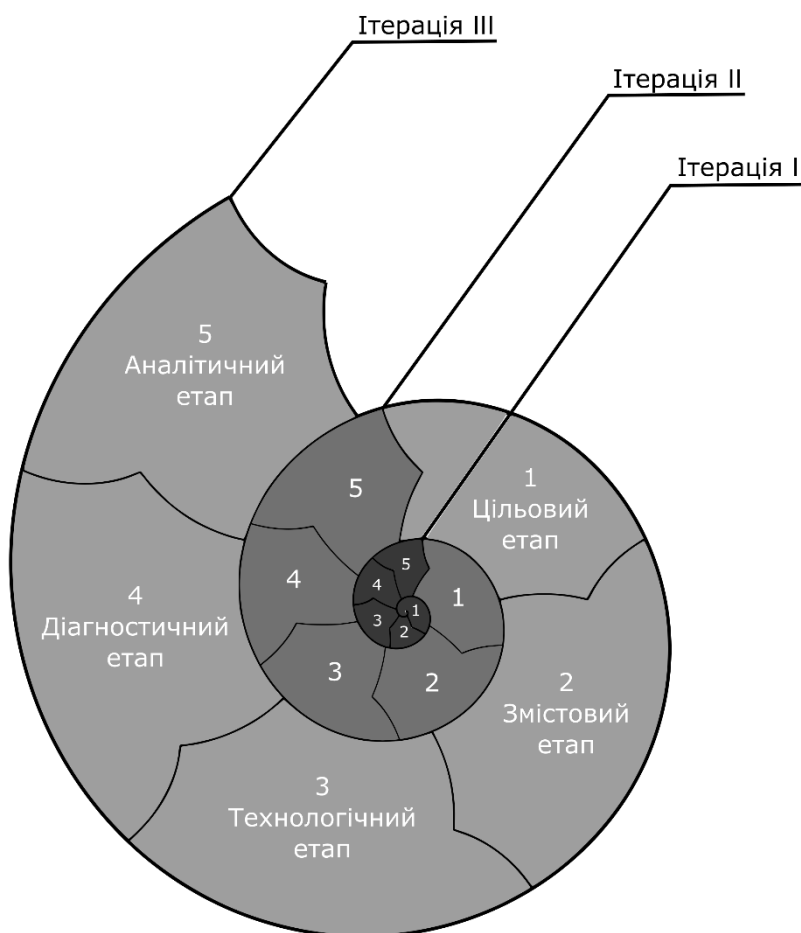


Рис. 2.1. Схематичне подання послідовності етапів проектування методики застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики

На основі проведеного аналізу результатів i -ї ітерації відбувається переосмислення, коригування та уточнення мети і завдань, поставлених на i -й ітерації, причому нові мета і завдання включають внесення змін на кожному з етапів $(i+1)$ -ї ітерації, якщо це необхідно. Ідею такої побудови моделі було запозичено з технологій розробки ПЗ, де аналогічним чином описується процес створення програмних засобів (спіральна модель життєвого циклу програмного засобу та ітеративна розробка) [17, с. 50].

Кожній ітерації відповідає новий виток спіралі (рис. 2.1). Спіраль починається в центрі схеми і, взагалі кажучи, має необмежену кількість витків.

У рамках освітнього процесу у ЗВО зручно асоціювати кожну нову ітерацію з новим навчальним роком, оскільки при такому підході щоразу

проходитиме повноцінний цикл з усіх п'яти етапів з урахуванням досвіду, набутого протягом чергового року.

Зміни, що вносяться між двома ітераціями, можуть як носити характер незначних виправлень і доповнень, так і мати більший масштаб (зміна методів і засобів навчання, зміна освітньої парадигми тощо). Метою внесення незначних змін є передусім вдосконалення курсу в межах загалом сталого напрямку розвитку, тимчасом як вагоміші зміни покликані змінити сам напрям розвитку. У випадку внесення значних змін знадобляться нові ітерації для повернення курсу до стану, за відпрацьованістю близького до попереднього варіанту.

Для оптимізації процесу вдосконалення курсу (зокрема мінімізації кількості ітерацій), незалежно від масштабу запланованих змін, варто регулярно фіксувати необхідні виправлення, доповнення, нові ідеї та нові актуальні питання, розгляд яких варто включити до курсу наступного року, і навпаки, питання, розгляд яких можна зробити менш детальним, винести на самостійне опрацювання чи не включати взагалі. У кінці навчального року, з урахуванням результатів етапу діагностування, а також усіх зауважень, самостійно занотованих викладачем у ході навчального процесу, здійснюється детальний аналіз та розпочинається нова ітерація процесу проектування методики. За умов успішної реалізації описаного підходу викладач має більше часу для попереднього обміркування змін до методики та їх системного внесення.

Для ефективної реалізації даної моделі вважаємо за доцільне здійснювати документування кожної ітерації. Оскільки документація використовуватиметься передусім самим викладачем (колективом викладачів), вона може мати неформальний характер. Головною вимогою до такої документації є її зручність передусім для окремого викладача (колективу викладачів). У додатку 3 подано приклади документації, використаної нами під час розробки і вдосконалення пропонованої методики.

Спираючись на послідовність етапів проектування (рис. 2.1), розроблено компоненти моделі застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики (рис. 2.2).

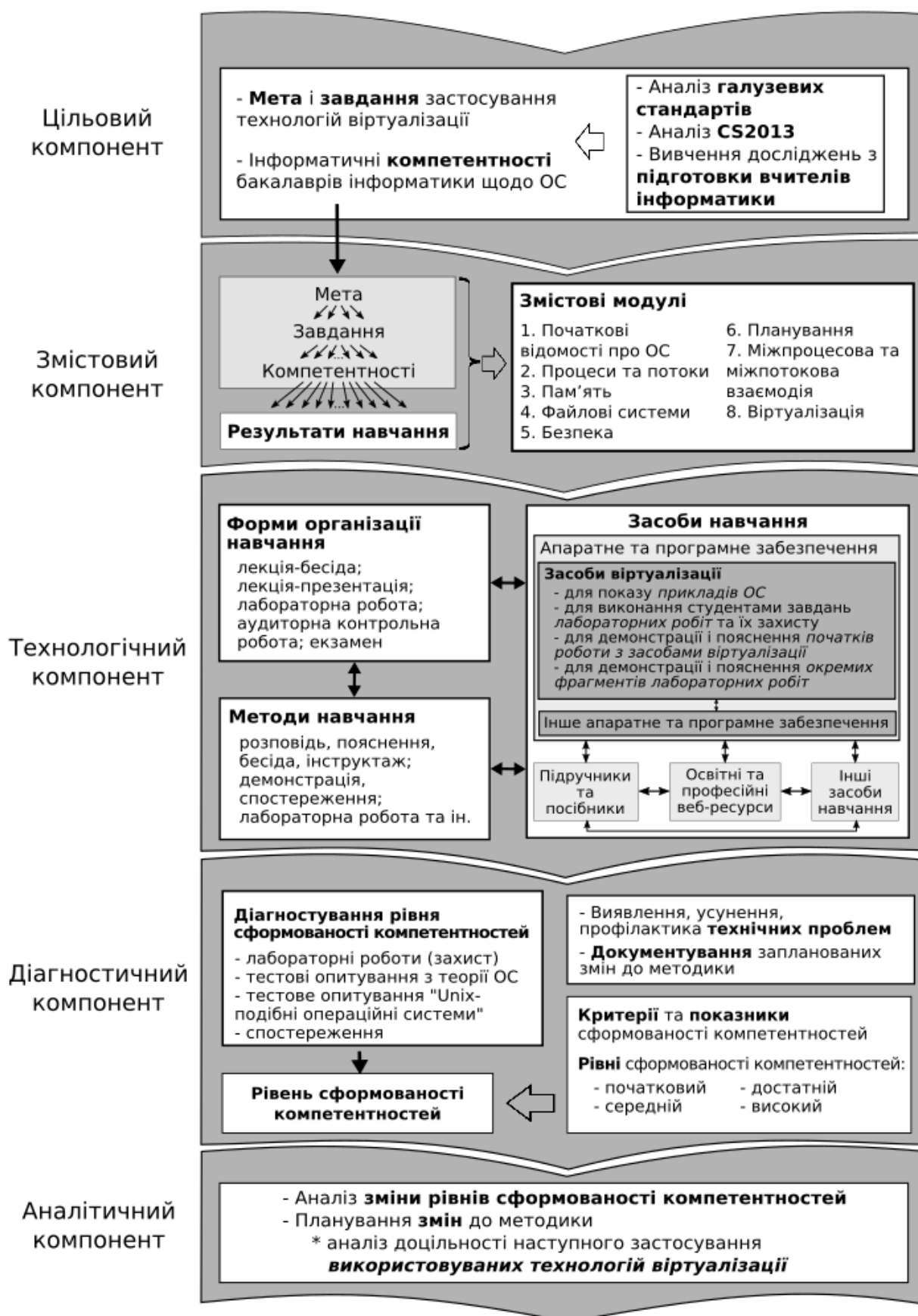


Рис. 2.2. Модель застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики

1. Цільовий компонент. Складові цільового компоненту зумовлено галузевим стандартом вищої освіти України за напрямом підготовки 6.040302 «Інформатика» [98; 99], проектом стандарту вищої освіти України для предметної спеціалізації 014.09 «Середня освіта (Інформатика)» для бакалаврського рівня освіти [100], проаналізованими дослідженнями з підготовки вчителів інформатики, світовим досвідом навчання ОС (керівними принципами розробки навчальних програм Computer Science Curricula 2013 [213], підручниками з ОС, відомостями з веб-ресурсів ЗВО).

В основу мети та завдань застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики покладено інформатичні компетентності бакалаврів інформатики щодо ОС, визначені у п. 1.2.2.

Мета застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики полягає у створенні умов для формування в студентів інформатичних компетентностей бакалаврів інформатики щодо ОС.

Відповідно до мети сформульовано *завдання, інформатичні компетентності* бакалаврів інформатики щодо ОС та *результати навчання*. Схематичне подання утвореної у такий спосіб чотирирівневої структури зображено на рис. 2.3.

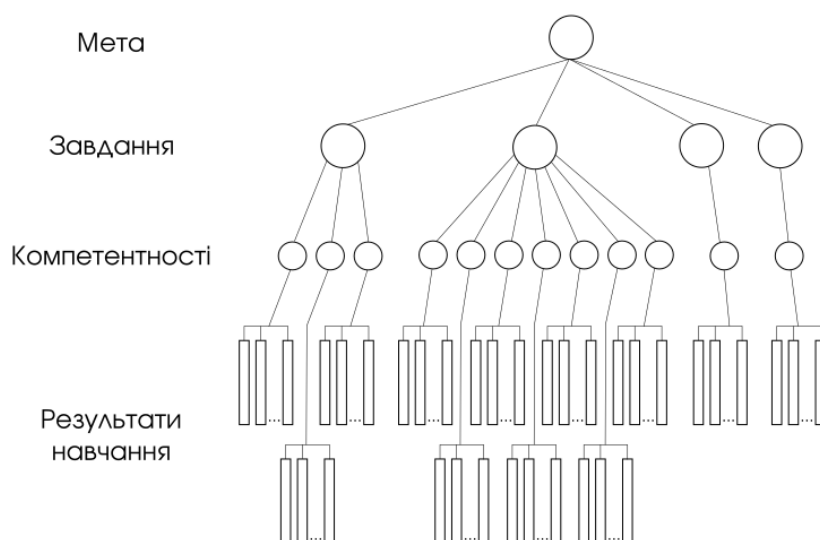


Рис. 2.3. Схематичне подання чотирирівневої структури цільового компоненту дидактичної моделі

Другий і третій рівні чотирирівневої структури наведено у табл. 2.1.

Таблиця 2.1

**Завдання застосування технологій віртуалізації Unix-подібних ОС
у підготовці бакалаврів інформатики та відповідні їм компетентності**

Завдання	Інформатичні компетентності з ОС
1) Навчання основ теорії ОС	- знати та розуміти основні етапи, напрями та тенденції розвитку ОС; - орієнтуватися у класифікації ОС, з урахуванням їх архітектури та сфери застосування; - знати та розуміти теоретичні основи будови та функціонування ОС
2) Освоєння основ адміністрування ОС	- вміти використовувати інтерфейс користувача Unix-подібних ОС (графічний інтерфейс і / або інтерфейс командного рядка); - вміти налаштовувати мережне з'єднання у Unix-подібних ОС; - вміти встановлювати програмне забезпечення у Unix-подібних ОС, виконувати його наступне оновлення та деінсталяцію; - вміти відстежувати роботу процесів та потоків в ОС і керувати ними; - вміти здійснювати моніторинг використання різних видів пам'яті; - володіти основами роботи з файловою системою у Unix-подібних ОС; - володіти основами роботи з налаштуваннями системи безпеки ОС
3) Навчання основ системного програмування	володіти основами системного програмування мовою C++ у Linux і / або Windows
4) Створення передумов для подальшої самоосвіти з ОС	вміти одержувати довідкові відомості щодо використання ОС та програмування у них

2. Змістовий компонент дидактичної моделі відповідає змісту навчальної дисципліни «Операційні системи та системне програмування», який, у свою чергу, формується згідно з метою та завданнями застосування технологій віртуалізації у підготовці бакалаврів інформатики, відповідними інформатичними компетентностями щодо ОС й очікуваними результатами навчання. Визначено змістові модулі з навчальної дисципліни «Операційні системи та системне програмування», для яких є педагогічно виваженим застосування технологій віртуалізації. Детальніший опис змістового компонента наведено у п. 3.2.

3. Технологічний компонент. Визначаються форми, методи та засоби, спрямовані на досягнення поставленої мети та завдань. Передбачено

застосування таких *форм навчання*, як лекція-бесіда, лекція-презентація, лабораторна робота, аудиторна контрольна робота, екзамен. Використовуються такі *методи навчання*, що належать до словесних (розповідь, пояснення, бесіда, інструктаж), наочних (демонстрація, спостереження), практичних (лабораторна робота) тощо. Серед передбачених моделлю і відповідною методикою *засобів навчання* виділимо апаратне та програмне забезпечення (у межах якого, у свою чергу, виокремимо **засоби віртуалізації**), підручники та посібники, освітні й професійні веб-ресурси, а також інші засоби навчання (наприклад, звичайні та маркерні дошки). Питання добору засобів віртуалізації досліджено у п. 2.3. Використання вибраних засобів віртуалізації у курсі з ОС застосування інших засобів навчання у цьому курсі, як і форм та методів навчання, описано у п. 3.2.

4. Діагностичний компонент. Здійснюється діагностування рівня сформованості визначених у межах цільового компоненту компетентностей шляхом перевірки окремих результатів навчання під час захисту лабораторних робіт, тестових опитувань з теорії ОС, тестового опитування «Unіх-подібні операційні системи». Детальніше про реалізацію вказаних форм контролю йдеться у розділі 3.

У межах діагностичного компоненту визначено *критерії, показники та рівні сформованості інформатичних компетентностей* бакалаврів інформатики щодо ОС.

Діагностування рівнів сформованості компетентностей здійснюється відповідно до наступних *критеріїв сформованості інформатичних компетентностей щодо ОС*.

- 1) Теоретичний (знання основ теорії ОС).
- 2) Адміністративний (володіння основами адміністрування ОС).
- 3) Програмістський (володіння основами адміністрування ОС).
- 4) Самонавчальний (здатність до наступної самоосвіти з ОС).

Показники сформованості інформатичних компетентностей щодо ОС виокремлено відповідно до мети та завдань методики, згідно з попередньо визначеними компетентностями й описано у п. 3.1.

Передбачено оцінювання за чотирма рівнями сформованості інформатичних компетентностей щодо ОС: початковий (2 бали), середній (3 бали), достатній (4 бали), високий (5 балів). Характеристики кожного рівня наведено у п. 3.2.

Також у межах діагностичного компоненту відбувається виявлення й усунення технічних проблем.

Практичний досвід застосування технологій віртуалізації під час навчання ОС пов'язаний з підвищеним ризиком виникнення програмних та апаратних збоїв під час занять. Збої можуть виникати передусім під час проведення лабораторних робіт й нерідко прямо чи опосередковано спричинені застосуванням засобів віртуалізації. Подібні проблеми ускладнюють та уповільнюють процес виконання студентом завдань лабораторної роботи, а у деяких випадках унеможливають продовження заняття для окремого студента чи всієї групи, доки несправність не буде усунута. У зв'язку з такою високою залежністю виконання лабораторних робіт від апаратно-програмних засобів вирішення технічних проблем нерідко необхідно шукати одразу по факту їх виникнення, тобто безпосередньо під час заняття.

Успішність ліквідації програмних та апаратних збоїв значною мірою залежить від досвіду викладача, який проводить заняття. Водночас потенційні технічні проблеми може бути враховано у процесі розробки методики, наприклад, шляхом усвідомленого добору ПЗ, включення до інструктивно-методичних матеріалів вказівок, покликаних допомогти студентам самостійно усувати найпоширеніші програмні й апаратні збої та здійснювати їх профілактику.

Однак жодний із запобіжних заходів не гарантує повної захищеності від програмних та апаратних збоїв. Це пояснюється зокрема теорією надійності ПЗ [92, с. 13]. Подібна ситуація тією чи іншою мірою притаманна багатьом інформатичним дисциплінам, однак курс з ОС належить до таких, що найбільш критично залежать від наявності чи відсутності технічних проблем у зв'язку зі складністю і часто – багаторівневістю ПЗ. У зв'язку з цим вважаємо регулярне

документування технічних та методичних проблем, ідей та запланованих змін до методики особливо важливим передусім у межах випробувального компоненту. У додатку 3 запропоновано приклад неформальної викладацької документації для фіксування зауважень до лекцій (лекційний блок) та лабораторних робіт (лабораторний блок).

5. Аналітичний компонент. Відбувається аналіз ходу й результатів реалізації методики протягом поточної ітерації. Окрім результатів діагностування рівня сформованості компетентностей, розгляду можуть підлягати власні спостереження викладачів, які здійснювали навчальний процес відповідно до запропонованої методики, враження студентів, думка запрошених експертів тощо.

Наступна ітерація процесу проектування методики передбачає корегування мети і завдань відповідно до підсумків аналізу. Зокрема, використовувані засоби віртуалізації може бути замінено або доповнено іншими, а методика їх застосування – вдосконаленою.

На основі побудованої моделі застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики розроблено методику застосування таких технологій, описану у розділі 3.

2.3. Критерії добору програмних засобів віртуалізації у навчанні бакалаврів інформатики Unix-подібних операційних систем

Застосування технологій віртуалізації вимагає спеціального програмного, а в низці випадків також і апаратного забезпечення. Про відповідне апаратне забезпечення ще йтиметься далі у цьому підрозділі, однак взаємодія користувача з віртуалізованою ОС здійснюється саме через програмне забезпечення, – програмні засоби віртуалізації, або віртуалізаційне програмне забезпечення (віртуалізаційне ПЗ).

Розробкою вимог до програмних засобів, що використовуються у навчальному процесі, займалися М. І. Жалдак [64; 65], В. Ю. Габрусев [28], В. В. Лапінський [65], Ю. І. Машбиць [95], В. М. Монахов [102], С. А. Раков

[141], С. О. Семеріков [151], Ю. В. Триус [173], М. І. Шут [65] та ін. У. П. Когут пропонує здійснювати вибір програмних засобів для фундаменталізації підготовки бакалаврів інформатики з інформатичних дисциплін за наступними критеріями: методична доцільність; інтуїтивно зрозумілий інтерфейс; україномовний інтерфейс; апаратна сумісність; програмна сумісність; ліцензійна чистота [78].

Зазначимо, що наявні засоби віртуалізації переважно призначені для реалізації не однієї конкретної задачі, а цілого спектру задач (див. п. 1.2). Це справедливо і для засобів віртуалізації, котрі застосовуються у галузі освіти. Вважаємо таку ситуацію сприятливою, оскільки вона надає додаткові можливості у контексті навчання протягом життя: у майбутньому студентові як випускнику і дипломованому фахівцю, цілком імовірно, знадобиться і використовувати віртуалізаційне ПЗ як засіб для самостійного вивчення нових ОС, і застосувати засоби віртуалізації за іншим призначенням, наприклад, для віртуалізації серверів чи віртуалізації робочого столу. Імовірно, що оптимальним за таких умов стало б деяке універсальне віртуалізаційне ПЗ, котрим випускник зміг би користуватися не лише на заняттях, а й у своїй майбутній професійній діяльності. На жаль, вибрати в ролі навчального засобу таке віртуалізаційне ПЗ, яке однаково добре підходило б для всіх інших можливих сфер застосування, неможливо у зв'язку з великим розмаїттям технологій віртуалізації, котрі суттєво відрізняються за методом реалізації та напрямом застосування (див. п. 1.2). Тож, формулюючи критерії добору програмного засобу віртуалізації, ми орієнтуватимемося передусім на навчальне призначення такого засобу, зважаючи, однак, і на потенційну можливість використання цього програмного продукту за іншими напрямками, а також на перспективу його подальшої підтримки і розвитку (умовно кажучи, які шанси, що цей засіб буде актуальний через п'ять років).

На основі поданих вище критеріїв, специфіки віртуалізаційного ПЗ порівняно з іншими програмними засобами, що використовуються у навчальному процесі, вивчення досвіду навчання Unix-подібних ОС у

вітчизняних та закордонних ЗВО, а також власного досвіду викладання дисципліни «Операційні системи та системне програмування» бакалаврам інформатики у Житомирському державному університеті імені Івана Франка, сформовано наступні **критерії добору програмних засобів віртуалізації у навчанні бакалаврів інформатики Unix-подібних ОС:**

- (1) відповідність використовуваної технології віртуалізації потребам курсу;
- (2) універсальність;
- (3) ліцензійна чистота;
- (4) функціональність інтерфейсу;
- (5) популярність та підтримка.

З метою з'ясування ступеню проявлення критеріїв у межах кожного з критеріїв виокремлено показники, які підлягають оцінюванню (рис. 2.4).

Розглянемо процес добору програмних засобів віртуалізації у навчанні бакалаврів інформатики Unix-подібних ОС за сформульованими критеріями та показниками. Зауважимо, що у процесі розробки критеріїв і показників та її наступного застосування було взято за основу узагальнену систематизацію технологій віртуалізації, описану у п. 1.1.

Загалом добір програмних засобів відбувався у два етапи.

I етап. Добір технологій віртуалізації за критерієм (1) «відповідність використовуваної технології віртуалізації потребам курсу».

- **Підготовча стадія.** Попередній відбір технологій віртуалізації за показниками (1.1.1) / (1.2.1), (1.2.2) (рис. 2.5).

- **Основна стадія.** Добір технологій віртуалізації за показниками (1.1.2)-(1.1.10), (1.2.2), (1.2.3) (рис. 2.6).

II етап. Добір засобів віртуалізації за критеріями (2)-(5) шляхом порівняльного аналізу безкоштовних та умовно безкоштовних програмних засобів, в основі яких лежать технології, вибрані на I етапі (рис. 2.7).

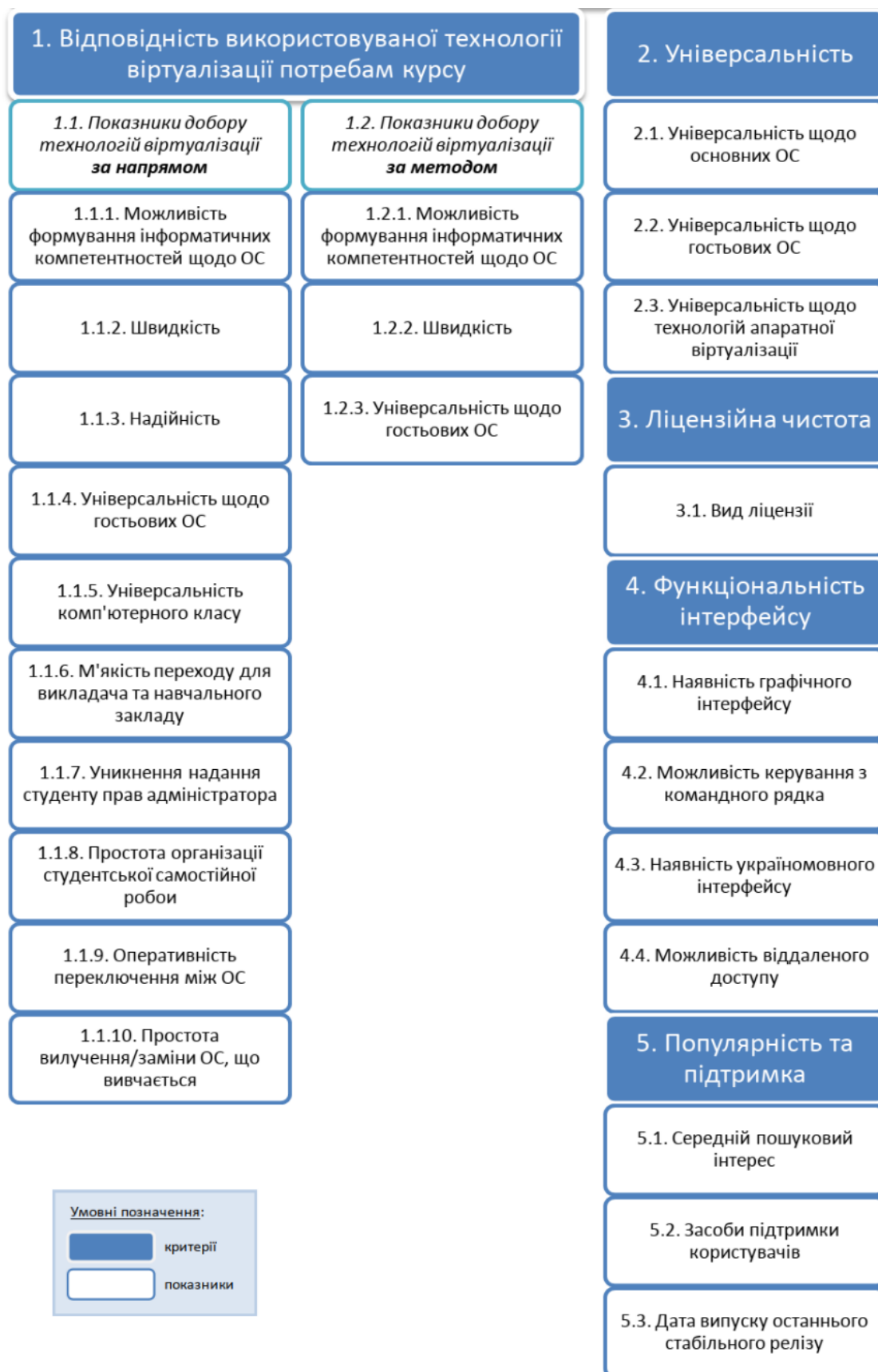


Рис. 2.4. Схематичне подання критеріїв добору програмних засобів віртуалізації у навчанні бакалаврів інформатики Unix-подібних ОС

Попередній відбір технологій віртуалізації



Рис. 2.5. Схема попереднього відбору технологій віртуалізації за напрямом віртуалізації (показник 1.1.1) та методом віртуалізації (показники 1.2.1, 1.2.2)

Добір технологій віртуалізації за критерієм (1)

Критерій 1. Відповідність використовуваної технології віртуалізації потребам курсу

1.1. Показники добору технологій віртуалізації **за напрямом**

- 1.1.1. Можливість формування інформатичних компетентностей щодо ОС
- 1.1.2. Швидкість
- 1.1.3. Надійність
- 1.1.4. Універсальність щодо гостьових ОС
- 1.1.5. Універсальність комп'ютерного класу
- 1.1.6. М'якість переходу для викладача та ЗВО
- 1.1.7. Уникнення надання студенту прав адміністратора
- 1.1.8. Простота організації студентської самостійної роботи
- 1.1.9. Оперативність переключення між ОС
- 1.1.10. Простота вилучення/заміни ОС, що вивчається

Набрана кількість балів суттєво більша за мінімальну?

так

ні

Гіпервізори I типу	7,7
Гіпервізори II типу	7,1
Гібридні гіпервізори	7,7
Віртуальні контейнери	7,3

Єдина ОС 3,0
Мультизавантаження 3,7

1.2. Показники добору технологій віртуалізації **за методом**

- 1.2.1. Можливість формування інформатичних компетентностей щодо ОС
- 1.2.2. Швидкість
- 1.2.3. Універсальність щодо гостьових ОС

Набрана кількість балів суттєво більша за мінімальну?

так

Динамічна трансляція	1,7
Паравіртуалізація	1,0

Рис. 2.6. Схема добору технологій віртуалізації за напрямом віртуалізації (показники 1.1.2-1.1.10) та методом віртуалізації (показники 1.2.2, 1.2.3)

Розглянемо процес добору програмних засобів віртуалізації Unix-подібних ОС у навчанні бакалаврів інформатики детальніше.

I етап. Перед власне добором програмних засобів віртуалізації задля зменшення кількості порівнюваних альтернатив здійснено попередній відбір технологій віртуалізації, на яких ґрунтуються ці засоби, за двома показниками **критерію (1) «відповідність використовуваної технології віртуалізації потребам курсу»:** показником (1.1.1) / (1.2.1) «можливість формування інформатичних компетентностей щодо ОС» та показником (1.2.2) «швидкість». Розглянемо обґрунтування потреби у такому відборі.

На момент розробки пропонованих критеріїв та їх застосування для добору засобів віртуалізації діяв галузевий стандарт вищої освіти України за напрямом підготовки 6.040302 «Інформатика» [98; 99].

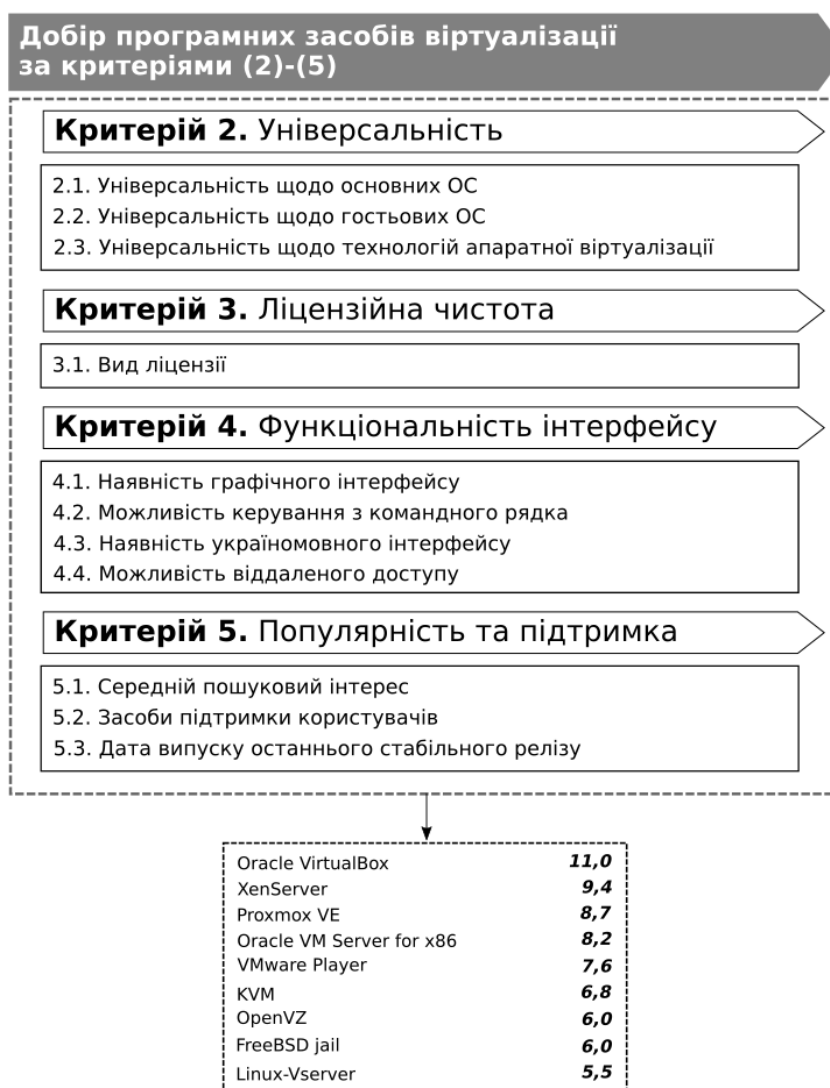


Рис. 2.7. Схема добору програмних засобів віртуалізації за критеріями (2)-(5)

Серед компетентностей, визначених цим галузевий стандартом (*умінь*, за термінологією стандарту), зазначено, що бакалавр інформатики має, зокрема, «володіти основними прийомами застосування системного програмного забезпечення інформаційної системи» (З.СВ.Д.01.ПР.Н.07), «вміти застосовувати основні функції операційної системи для роботи з файлами та каталогами» (З.СВ.Д.01.ПР.Р.11), «вміти застосовувати сервісне програмне забезпечення (програми-архіватори, антивірусні програми, програми-утиліти тощо)» (З.СВ.Д.01.ПР.Р.13), «вміти інсталювати програмні засоби» (З.СВ.Д.01.ПР.Р.14) [98]. Вказані компетентності передбачають роботу з ОС у цілому, а не лише з окремими програмами, причому кожен студент повинен працювати з власним екземпляром ОС, аби бачити ефект від своїх дій, а також мати адміністративні права у межах цього екземпляру.

За новим Законом України «Про вищу освіту» [22] передбачено розробку нового стандарту вищої освіти для спеціальності 014.09 «*Середня освіта (Інформатика)*», яка прийшла на зміну напряму підготовки 6.040302 «*Інформатика*». Однак на момент завершення даного дослідження такий стандарт не затверджено, а проєкті стандарту вищої освіти України для предметної спеціалізації 014.09 «*Середня освіта (Інформатика)*» для бакалаврського рівня освіти (2017 рік) [100] компетентності, пов'язані з ОС, визначено досить загально. Водночас, під час розробки переліку компетентностей, які підлягають формуванню відповідно до методики застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики, враховано як попередній галузевий стандарт для напряму підготовки 6.040302 «*Інформатика*» [98; 99], так і проєкт нового стандарту [100], компетентності майбутніх учителів інформатики, визначені О. М. Спіріним [158], закордонний досвід навчання ОС (зокрема рекомендації [213]). Тому вважаємо за доцільне спиратися на розроблений перелік компетентностей (п. 2.2) під час оцінювання програмних засобів віртуалізації за показником (1.1.1 / 1.2.1).

Такі *інформатичні компетентності* бакалаврів інформатики щодо ОС, як «вміти налаштовувати мережне з'єднання у Unix-подібних ОС», «вміти встановлювати програмне забезпечення у Unix-подібних ОС, виконувати його наступне оновлення та деінсталяцію», «вміти відстежувати роботу процесів та потоків в ОС і керувати ними», «вміти здійснювати моніторинг використання різних видів пам'яті», «володіти основами роботи з файловою системою у Unix-подібних ОС», «володіти основами роботи з налаштуваннями системи безпеки ОС» також повністю чи частково передбачають роботу студента із власним екземпляром ОС та наявність адміністративних прав у межах цього екземпляру.

Отже, для формування визначених компетентностей не підійдуть засоби віртуалізації на основі напрямів та методів, які не надають повнофункціональної ОС з правами адміністратора для кожного студента (віртуальні програмні застосунки, сервіси віддаленого робочого столу, емуляція системних бібліотек). Також відсіюванню підлягають засоби віртуалізації на основі методів, котрі не забезпечують швидкості, достатньої для роботи сучасної, не навчальної, ОС (повна емуляція).

Емуляція системних бібліотек віртуалізує окремі програми, що не дозволяє формувати перелічені вище уміння. Повна емуляція (застосована безпосередньо) є суттєво повільнішою за решту технологій [168, с. 98-99], а засоби пришвидшення її роботи сумісні з невеликою кількістю ОС [272], до того ж, навчання бакалаврів інформатики дисципліни «Операційні системи та системне програмування» не потребує імітації всієї архітектури комп'ютера. Втім, повна емуляція часто є складовою інших технологій, тому в даному випадку йдеться лише про відмову від неї в чистому вигляді.

Віртуалізація програмних застосунків дає можливість працювати не з усією ОС, а з окремими програмами. Віртуалізація робочого столу на базі сервісів віддаленого робочого столу (remote desktop service, RDS) надає всім користувачам паралельний віддалений доступ до спільного екземпляра ОС. Засоби інфраструктури віддаленого робочого столу (virtual desktop infrastructure, VDI) виділяють окремий екземпляр ОС із можливістю надати користувачу

адміністративні повноваження, однак зазвичай ідеться про віртуальні робочі столи, створені за допомогою інших засобів віртуалізації (зокрема гіпервізорів), тому задля уникнення дублювання наразі виключаємо VDI-засоби з порівняння. Водночас, технології віртуалізації робочого столу продовжують розвиватися, передусім у зв'язку зі зростанням популярності хмарних технологій, тож згодом таке рішення може потребувати перегляду.

Таким чином, після попереднього відбору (рис. 2.5) множину альтернатив було звужено до наступних технологій віртуалізації: гіпервізори I типу, гіпервізори II типу, гібридні гіпервізори, віртуальні контейнери (за напрямом); динамічна трансляція, паравіртуалізація (за методом).

Окрім того, можливе також навчання Unix-подібних ОС без застосування технологій віртуалізації. Ідеться про наявність комп'ютерного класу, в якому Unix-подібну ОС, що вивчається, встановлено на реальні машини – або як єдину ОС на кожному із цих комп'ютерів, або поряд із іншою ОС, найчастіше в окремий розділ жорсткого диску (так зване мультизавантаження, англ. multi-booting). Таким чином, одержуємо наступні *варіанти навчання Unix-подібних ОС*: **єдина ОС, мультизавантаження, гіпервізори I типу, гіпервізори II типу, гібридні гіпервізори, віртуальні контейнери.**

Зауважимо, що всі перелічені варіанти навчання може бути використано і для віртуалізації Unix-подібних ОС, що є пріоритетним у межах даного дослідження, і для інших ОС. Водночас, віртуальні контейнери не підходять для віртуалізації ОС, які не належать до Unix-подібних (наприклад, Windows).

Оцінимо визначені варіанти навчання за рештою показників групи (1.1) та здійснимо порівняльний аналіз варіантів навчання Unix-подібних ОС у підготовці бакалаврів інформатики у контексті навчання дисципліни «Операційні системи та системне програмування».

Показник (1.1.2) «швидкість» визначається відсутністю явних затримок у роботі Unix-подібної ОС, що вивчається, й оцінюється за параметрами:

0 балів – ОС, що вивчається, працює дуже повільно, і це суттєво впливає на продуктивність виконання студентами завдань лабораторної роботи;

1 бал – ОС, що вивчається, працює помітно повільніше, ніж це потенційно можливо на даному обладнанні, однак вплив затримок на продуктивність виконання студентами завдань лабораторної роботи є помірним;

2 бали – ОС, що вивчається, працює дещо повільніше, ніж це потенційно можливо на даному обладнанні, однак вплив на продуктивність виконання студентами завдань лабораторної роботи є незначним;

3 бали – ОС, що вивчається, працює швидко, але повільніше, ніж це потенційно можливо на даному обладнанні; вплив затримок на продуктивність виконання студентами завдань лабораторної роботи мінімальний або відсутній;

4 бали – ОС, що вивчається, працює з максимальною швидкістю, доступною на даному обладнанні.

Показник (1.1.3) «надійність» пов'язаний з відсутністю збоїв протягом деякого періоду часу. Оцінювання даного показника здійснюється згідно з параметрами:

0 балів – надійність роботи ОС, що вивчається, є низькою, збої трапляються дуже часто;

1 бал – надійність роботи ОС, що вивчається, є середньою, збої трапляються досить часто, однак пов'язані з ними ризики можуть бути компенсовані, наприклад, шляхом збереження проміжних станів віртуалізованої ОС задля їх наступного використання як резервних копій у разі збою;

2 бали – надійність роботи ОС, що вивчається, є високою, збої трапляються досить рідко, однак дещо частіше, ніж коли б ОС не було віртуалізовано;

3 бали – надійність роботи ОС, що вивчається, є найвищою, збої трапляються так само рідко, як коли б ОС не було віртуалізовано.

Показник (1.1.4) «універсальність щодо гостей ОС» означає підтримку широкого спектру ОС у ролі гостьової. Для оцінювання показника (1.1.4) використовуються параметри:

0 балів – низька універсальність (оцінюваний варіант дозволяє навчати лише однієї ОС, наприклад, лише FreeBSD або лише Linux);

1 бал – середня універсальність (оцінюваний варіант дозволяє навчати лише ОС однієї платформи, наприклад, лише Unix-подібних);

2 бали – висока універсальність (оцінюваний варіант дозволяє навчати ОС різних платформ, у тому числі Unix-подібних, однак перелік потенційно доступних ОС є невеликим й обмежується кількома ОС кожної платформи);

3 бали – висока універсальність (оцінюваний варіант дозволяє навчати ОС різних платформ, у тому числі Unix-подібних, а перелік потенційно доступних ОС є великим).

Показник (1.1.5) «універсальність комп'ютерного класу» визначається можливістю проведення у даному комп'ютерному класі занять із інших дисциплін й оцінюється відповідно до параметрів:

0 балів – низька універсальність комп'ютерного класу (комп'ютерний клас адаптований для проведення занять з «Операційних систем та системного програмування», тимчасом як для організації у цьому класі занять з інших дисциплін потрібні значні зусилля);

1 бал – середня універсальність комп'ютерного класу (комп'ютерний клас однаково добре підходить як для проведення занять з «Операційних систем та системного програмування», так і для навчання інших дисциплін, але, можливо, під час занять знадобиться використання засобів віртуалізації чи перезавантаження комп'ютерів для переходу з однієї ОС в іншу);

2 бали – висока універсальність комп'ютерного класу (комп'ютерний клас адаптований для проведення занять з інших дисциплін, тимчасом як для організації у цьому класі занять з «Операційних систем та системного програмування» достатньо веб-переглядача та мережного з'єднання).

Показник (1.1.6) «м'якість переходу для викладача та ЗВО» тотожна можливості розпочати використання даного варіанту навчання Unix-подібних ОС без кардинальних змін у програмному та апаратному забезпеченні комп'ютерного класу чи навчального закладу загалом. Даний показник оцінюється за допомогою наступних параметрів:

0 балів – низька м'якість переходу (радикальні зміни в програмному і/або

апаратному забезпеченні);

1 бал – середня м'якість переходу (суттєві зміни у програмному і/або апаратному забезпеченні);

2 бали – висока м'якість переходу (незначні зміни у програмному і/або апаратному забезпеченні).

Показник (1.1.7) «уникнення надання студенту прав адміністратора» має на меті забезпечення стабільної роботи комп'ютерного класу і передбачає надання переваги варіантам, що дозволяють не виділяти студенту адміністративних прав у межах комп'ютера, за яким працює цей студент. Для оцінювання даного показника використовуються параметри:

0 балів – немає можливості уникнути надання студенту прав адміністратора у межах комп'ютера, за яким працює студент;

1 бал – є можливість уникнути надання студенту прав адміністратора у межах комп'ютера, за яким працює студент.

Показник (1.1.8) «простота організації студентської самостійної роботи» означає потенційну доступність для студента роботи з Unix-подібною ОС, що вивчається, у позаурочний час, зокрема на власному комп'ютері. Оцінювання показника здійснюється відповідно до параметрів:

0 балів – для самостійної роботи з ОС, що вивчається, необхідні значні організаційні зусилля студента (потрапити після занять у певний комп'ютерний клас; встановити ОС, що вивчається, на власний комп'ютер як єдину чи як другу ОС; самостійно освоїти програмний засіб віртуалізації за умови відсутності досвіду роботи з аналогічними засобами під час занять тощо);

1 бал – для самостійної роботи з ОС, що вивчається, потрібні незначні організаційні зусилля студента (встановити на власний комп'ютер програмний засіб віртуалізації, тотожний чи аналогічний тому, який використовується під час занять; мати веб-переглядач, мережне з'єднання та перебувати в зоні дії локальної мережі навчального закладу; мати веб-переглядач, мережне з'єднання, включно з доступом до Інтернету тощо).

Показник (1.1.9) «оперативність переключення між ОС» передбачає

підтримку швидкого переходу від однієї ОС до іншої за умови одночасної наявності на одному комп'ютері декількох ОС, у тому числі віртуалізованих. Параметри для оцінювання даного показника є наступними:

0 балів – для переключення між різними ОС потрібне перезавантаження комп'ютера;

1 бал – для переключення між різними ОС перезавантаження комп'ютера не потрібне.

Показник (1.1.10) «простота вилучення / заміни ОС, що вивчається» дає змогу за мінімальних затрат часу та зусиль перейти на вивчення іншої ОС, здійснювати одночасне вивчення кількох ОС тощо. Показник оцінюється згідно з параметрами:

0 балів – вилучення/заміна ОС, що вивчається, є простими;

1 бал – вилучення/заміна ОС, що вивчається, є складнішими і можуть передбачати вирішення питань з наступним використанням звільнених розділів жорсткого диску, перевстановлення ПЗ, необхідного для проведення занять з інших дисциплін, тощо.

Для кожного з варіантів навчання обчислимо суми нормованих оцінок за формулою

$$S = \sum_{i=1}^n \frac{x_i}{x_i^{max}}$$

де n – кількість показників, x_i – оцінка за i -м показником, x_i^{max} – максимально можливе значення оцінки за i -м показником.

Результати оцінювання варіантів навчання Unix-подібних ОС за показниками групи (1.1) наведено у табл. 2.2.

Помітно менша сума балів відповідає варіантам, котрі не передбачають віртуалізації, – єдина ОС (3,000 бала) та мультизавантаження (3,667 бала). Суми балів, одержані для решти варіантів, які ґрунтуються на технологіях віртуалізації, близькі за значенням (від 7,083 до 7,667 бала), причому гіпервізори I типу та гібридні гіпервізори все ж дещо переважають віртуальні контейнери, а віртуальні контейнери – гіпервізори II типу.

Таблиця 2.2

Порівняння напрямів віртуалізації Unix-подібних ОС та альтернативних їм варіантів без віртуалізації за визначеними для них показниками групи (1.1) критерію (1)

Варіант навчання Unix-подібних ОС	Показники та діапазони можливих значень параметрів									
	Швидкість (0..4)	Надійність (0..3)	Універсальність щодо гостей ОС (0..3)	Універсальність комп'ютерного класу (0..2)	М'якість переходу для викладача та ЗВО (0..2)	Уникнення надання студенту прав адміністратора (0..1)	Простота організації студентської самостійної роботи (0..1)	Оперативність перемикання між ОС (0..1)	Простота вилучення/заміни ОС, що вивчається (0..1)	Сума нормованих оцінок
Єдина ОС	4	3	3	0	0	0	0	0	0	3,000
Мультизавантаження	4	2	3	1	1	0	0	0	0	3,667
Гіпервізор I типу	2	2	3	2	1	1	1	1	1	7,667
Гібридний гіпервізор	2	2	3	2	1	1	1	1	1	7,667
Віртуальні контейнери	3	2	1	2	1	1	1	1	1	7,250
Гіпервізор II типу	1	1	3	1	2	1	1	1	1	7,083

Враховуючи отримані результати порівняльного аналізу, надалі у цій роботі ми розглядатимемо лише варіанти навчання, які передбачають віртуалізацію – **гіпервізори I типу, гіпервізори II типу, гібридні гіпервізори, віртуальні контейнери.**

Перейдемо до порівняння методів віртуалізації Unix-подібних ОС за показниками групи (1.2) критерію (1). Як і під час порівняння за показниками групи (1.1) цього критерію, спершу здійснимо відсіювання за показником (1.2.1). Ми вважаємо за доцільне застосування ПЗ, що ґрунтується на таких методах програмної віртуалізації, як *динамічна трансляція* та *паравіртуалізація*. Підтримка *апаратної віртуалізації* безпосередньо не впливає на можливість формування інформатичних компетентностей щодо ОС, а тому винесена до показника (2.3).

Після попереднього відбору за показником (1.2.1) і (1.2.2) було одержано такі *методи віртуалізації*, як **динамічна трансляція та паравіртуалізація.**

Зауважимо, що обидва методи підходять для віртуалізації Unix-подібних ОС, пріоритетною у межах даного дослідження. Проте якщо курсом з ОС передбачено також навчання ОС, які не належать до Unix-подібних (наприклад, Windows), то важливо враховувати, що метод паравіртуалізації у такому разі використати не можна.

Оцінимо відібрані методи за рештою показників групи (1.2) критерію (1) та виконаємо їх порівняльний аналіз.

Показник (1.2.2) «швидкість роботи» оцінюється за тими самими параметрами, що й аналогічний показник (1.1.2) групи (1.1).

Показник (1.2.3) «універсальність щодо гостей ОС» передбачає підтримку різноманітних ОС у ролі гостей. Для оцінювання показника використовуються параметри:

0 балів – низька універсальність (оцінюваний метод дозволяє використовувати у ролі гостьової лише ОС з відкритим кодом);

1 бал – висока універсальність (оцінюваний метод дозволяє використовувати у ролі гостьової ОС як з відкритим, так і з закритим кодом).

Результати оцінювання відібраних методів віртуалізації за показниками групи (1.2) наведено у (табл. 2.3).

З табл. 2.3 видно, що обидва методи віртуалізації одержали близькі за значенням суми балів (1,667 та 1). Надалі ми розглядатимемо ПЗ на базі обох цих методів (**динамічної трансляції** та **паравіртуалізації**) чи їх комбінації.

Таблиця 2.3

Порівняння методів віртуалізації Unix-подібних ОС за визначеними для них показниками групи (1.2) критерію (1)

Метод віртуалізації	Показники та діапазони можливих значень параметрів		
	Швидкість (0..3)	Універсальність щодо гостей ОС (0..1)	Сума нормованих оцінок
Динамічна трансляція	2	1	1,667
Паравіртуалізація	3	0	1

Поєднавши результати порівняльного аналізу, з одного боку, технологій віртуалізації та альтернативних їм варіантів навчання ОС за показниками групи (1.1), а з іншого – методів віртуалізації за показниками групи (1.2), визначимо **параметри для підсумкового оцінювання ПЗ за показниками критерію (1):**

0 балів – гіпервізор II типу (динамічна трансляція і/або паравіртуалізація); віртуальні контейнери;

1 бал – гіпервізор I типу (або динамічна трансляція, або паравіртуалізація); гібридний гіпервізор (або динамічна трансляція, або паравіртуалізація); гіпервізор II типу (або динамічна трансляція, або паравіртуалізація) у поєднанні з віртуальними контейнерами;

2 бали – гіпервізор I типу (і динамічна трансляція, і паравіртуалізація); гібридний гіпервізор (і динамічна трансляція, і паравіртуалізація); гіпервізор II типу (і динамічна трансляція, і паравіртуалізація) у поєднанні з віртуальними контейнерами.

Перейдемо до оцінювання **програмних засобів віртуалізації**, що ґрунтуються на виокремлених технологіях, за показниками критеріїв (2)-(5).

Критерій (2) «універсальність» складають показники:

(2.1) універсальність щодо основних ОС;

(2.2) універсальність щодо гостьових ОС;

(2.3) універсальність щодо технологій апаратної віртуалізації.

Показник (2.1) «універсальність щодо основних ОС» означає можливість встановлення віртуалізаційного ПЗ на комп'ютери з різними основними ОС, у тому числі ОС різних платформ. Показник оцінюється за наступними параметрами:

0 балів – підтримується лише одна основна ОС (або одна платформа ОС, наприклад, лише Linux, лише Windows, лише FreeBSD);

1 бал – підтримується декілька основних ОС (і ці ОС належать до різних платформ, наприклад, і Windows, і Linux); не потребує основної ОС (гіпервізори I типу, гібридні гіпервізори).

Показник (2.2) «універсальність щодо гостьових ОС» відрізняється від подібного показника (1.2.3) критерію (1) й спрямований передусім на виявлення підтримуваних платформ ОС. Один програмний засіб віртуалізації нерідко поєднує в собі декілька методів віртуалізації, а тому у переліках підтримуваних ОС можуть бути як ОС із відкритим, так і ОС із закритим кодом. Оцінювання показника здійснюється згідно з наступними параметрами:

0 балів – підтримується лише одна гостьова ОС;

1 бал – підтримуються гостьові ОС лише однієї платформи (наприклад, лише Linux, лише Unix-подібні); підтримуються різні гостьові ОС, у тому числі різних платформ, однак серед офіційно підтримуваних гостьових ОС немає безкоштовних;

2 бали – підтримуються різні гостьові ОС, у тому числі різних платформ, і серед офіційно підтримуваних гостьових ОС є безкоштовні.

Показник (2.3) «універсальність щодо технологій апаратної віртуалізації» визначається наявністю чи відсутністю підтримки програмним засобом процесорів з апаратною віртуалізацією (технології Intel VT-x чи AMD-V). Наявність такої підтримки дає змогу поєднати технології програмної та апаратної віртуалізації, чим потенційно пришвидшити роботу віртуалізованої ОС. Однак ряд програмних засобів не лише підтримують апаратну віртуалізацію, а й висувають її як обов'язкову системну вимогу, що робить таке ПЗ незастосовним на інших процесорах. Оцінювання показника (2.3) здійснюватимемо відповідно до параметрів:

0 балів – апаратна віртуалізація не підтримується;

1 бал – апаратна віртуалізація підтримується і є обов'язковою для всіх гостьових ОС;

2 бали – апаратна віртуалізація підтримується і є обов'язковою для частини гостьових ОС (наприклад, для ОС із закритим кодом);

3 бали – апаратна віртуалізація підтримується, але не є обов'язковою для жодної з гостьових ОС.

Критерій (3) «ліцензійна чистота» передбачає фінансову й організаційну можливість для навчального закладу використовувати програмний засіб згідно з ліцензією і характеризується єдиним показником **(3.1) «вид ліцензії»**. У реаліях вітчизняного навчального процесу одним із найпростіших та найдешевших є варіант застосування умовно безкоштовного та вільно поширюваного ПЗ (останнє надає більше додаткових можливостей, оскільки передбачає доступ до програмних кодів засобу), тому надалі ми зосередимось саме на такому ПЗ. Показник оцінюється за параметрами:

0 балів – власницьке ПЗ, використання платне;

1 бал – власницьке ПЗ, використання умовно безкоштовне (використання для навчальних закладів обмежене закладами-членами партнерських програм);

2 бали – власницьке ПЗ, використання умовно безкоштовне (використання для всіх навчальних закладів безкоштовне);

3 бали – вільно поширюване ПЗ; безкоштовне ПЗ.

Критерій (4) «функціональність інтерфейсу» має такі показники, як

(4.1) наявність графічного інтерфейсу;

(4.2) можливість керування з командного рядка;

(4.3) наявність україномовного інтерфейсу;

(4.4) можливість віддаленого доступу.

Показник (4.1) «наявність графічного інтерфейсу». Графічний інтерфейс спрощує роботу з програмним засобом для користувача-початківця. Оцінювання показника здійснюється за параметрами:

0 балів – немає графічного інтерфейсу;

1 бал – є графічний інтерфейс (можливо, сторонніх розробників).

Показник (4.2) «можливість керування із командного рядка». Керуваність із командного рядка уможливорює написання сценаріїв та утиліт для часткової автоматизації процесу проведення лабораторних робіт. Показник оцінюється відповідно до параметрів:

0 балів – немає можливості керування ПЗ з командного рядка;

1 бал – є можливість керування ПЗ з командного рядка (у тому числі

застосовуючи ПЗ сторонніх розробників).

Показник (4.3) «наявність україномовного інтерфейсу». Доступність україномовного інтерфейсу сприяє оволодінню україномовною термінологією. Оцінювання показника виконується згідно з параметрами:

0 балів – україномовного інтерфейсу немає;

1 бал – україномовний інтерфейс є, однак українська локалізація інтерфейсу часткова, і/або переклад містить численні неточності;

2 бали – україномовний інтерфейс є, в цілому українська локалізація інтерфейсу повна, а переклад – точний.

Показник (4.4) «можливість віддаленого доступу» передбачає наявність сумісних із даним ПЗ веб-інтерфейсів для роботи з віртуальними машинами (віртуальними контейнерами) через мережу. Доступність таких інтерфейсів відкриває перспективи для розміщення віртуалізованих ОС на локальному сервері начального закладу, на самостійно обслуговуваному чи орендованому глобального сервері тощо. Це може передбачати деякі додаткові організаційні зусилля та матеріальні затрати на початковому етапі, однак несе і ряд переваг. Це, зокрема, можливість доступу до віртуальних машин (контейнерів) з різних комп'ютерних класів чи з особистих комп'ютерів студентів за умови наявності підключення до мережі інтернет, причому у випадку глобального сервера такий доступ є й поза межами начального закладу. Іншою перевагою віддаленого розміщення віртуалізованих ОС є переведення витрат щодо модернізації із площини оновлення техніки в комп'ютерних класах у площину модернізації власного сервера чи оплати орендованого сервера. Досвід застосування технологій віддаленого доступу до віртуальних машин з навчальною метою описано зокрема у [5; 32]. Втім, реалізація потенціалу веб-інтерфейсів для віртуалізаційного ПЗ потребує готовності викладача і навчального закладу.

Показник (4.4) оцінюється відповідно до параметрів:

0 балів – веб-інтерфейсу для даного ПЗ немає;

1 бал – веб-інтерфейс для даного ПЗ є (можливо, веб-інтерфейс стороннього розробника).

Критерій (5) «популярність та підтримка» складають показники:

- (5.1) середній пошуковий інтерес;
- (5.2) засоби підтримки користувачів;
- (5.3) дата випуску останнього стабільного релізу.

Популярний продукт з більшою імовірністю стане в пригоді студенту в його майбутній професійній діяльності, має вищі шанси на тривалу підтримку. Оцінювання популярності програмних засобів віртуалізації ускладнене відсутністю авторитетних рейтингів такого ПЗ. Втім, наближеним аналогом популярності того чи іншого ПЗ може вважатися інтерес до цього ПЗ у всесвітній мережі. Для оцінювання інтересу до ПЗ ми застосували систему Google Тренди, в основі роботи якої лежить визначення частки пошукових запитів, котрі містили досліджуваний термін (у даному випадку – назви засобу віртуалізації). Для цього обчислюється відношення кількості пошукових запитів Google, до складу яких входив цей термін, за певний період часу, до загальної кількості пошукових запитів Google за той самий період. Одержане значення нормалізується (діапазон від 0 до 100). Також знаходяться середні значення нормалізованого відношення за весь розглянутий період часу [190]. Для зручності назовемо ці значення *середнім пошуковим інтересом (СПІ)*. З метою оцінювання популярності програмних засобів віртуалізації нами було використано СПІ, обчислений для діапазону з 2004 по 2014 роки. СПІ є **показником (5.1) «середній пошуковий інтерес»** критерію (5) «популярність та підтримка». Після обчислення значення СПІ здійснюється його оцінювання за параметрами:

0 балів – $0 \leq СПІ < 10$;

1 бал – $10 \leq СПІ < 30$;

2 бали – $30 \leq СПІ < 50$;

3 бали – $СПІ \geq 50$.

Показник **(5.2) «засоби підтримки користувачів»**. Наявність різноманітних форм підтримки користувачів (користувацька та технічна документація, форуми, списки розсилки тощо) допомагає в освоєнні програмного засобу та у разі

виникнення технічних проблем. Показник оцінюється за наступними параметрами:

0 балів – підтримка користувачів платна;

1 бал – підтримка користувачів переважно платна, однак є мінімальний обсяг безкоштовної підтримки (наявної в одній-двох формах); підтримка користувачів безкоштовна, однак дуже обмежена (наявна в одній-двох формах)

2 бали – підтримка користувачів переважно або повністю безкоштовна, а також наявна у багатьох формах.

Показник **(5.3) «дата випуску останнього стабільного релізу»** дає наближене уявлення про актуальність програмного забезпечення. Для оцінювання даного показника використовуються параметри:

0 балів – останній стабільний реліз випущено до 2007 року;

1 бал – останній стабільний реліз випущено з 2007-го по 2012-й рік включно;

2 бали – останній стабільний реліз випущено після 2013-го року.

На основі розглянутих критеріїв та показників добору проведемо порівняльний аналіз віртуалізаційного ПЗ, яке належить до категорій вільно поширюваного та умовно безкоштовного ПЗ.

У таблиці И.1 (додаток И) щодо кожного показника наведено фактичні відомості, взяті передусім з офіційних сайтів ПЗ, тимчасом як табл. 2.4 містить чисельні результати оцінювання згідно з показниками, а рис. 2.8 – візуалізацію цих результатів.

Важливо врахувати, що під час порівняльного аналізу ми спиралися на припущення про однакову вагу всіх показників. У кожному окремому випадку показники можуть мати різну вагу, і тоді, ймовірно, матимуть місце інші результати. У зв'язку з цим вбачаємо необхідність створення спеціальних засобів підтримки викладача у виборі віртуалізаційного ПЗ (довідкових матеріалів, інфографіки, експертних систем тощо), що давали б змогу викладачу індивідуалізувати вибір віртуалізаційного ПЗ відповідно до наявних потреб, можливостей та пріоритетів.

Таблиця 2.4

Порівняльний аналіз програмних засобів віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики: чисельні результати оцінювання

Назва ПЗ	Оцінки за критеріями та показниками; параметри оцінювання												Сума нормованих оцінок
	(1)	(2)			(3)	(4)				(5)			
	(1.1) / (1.2) 0..2	(2.1) 0..1	(2.2) 0..2	(2.3) 0..3	(3.1) 0..3	(4.1) 0..1	(4.2) 0..1	(4.3) 0..2	(4.4) 0..1	(5.1) 0..4	(5.2) 0..2	(5.3) 0..2	
Oracle VirtualBox	0	1	2	3	3	1	1	2	1	4	2	2	11.000
XenServer	1	1	2	2	3	1	1	0	1	1	2	2	9.417
Proxmox VE	2	0	2	2	3	1	1	0	1	0	2	2	8.667
Oracle VM Server for x86	1	1	1	2	3	1	1	0	1	0	1	2	8.167
VMware Player	0	1	2	3	1	1	1	0	0	1	2	2	7.583
KVM	0	0	2	1	3	1	1	0	1	0	2	1	6.833
OpenVZ	0	0	0	0	3	1	1	0	1	0	2	2	6.000
FreeBSD jail	0	0	0	0	3	1	1	0	1	0	2	2	6.000
Linux-Vserver	0	0	0	0	3	1	1	0	1	0	2	1	5.500

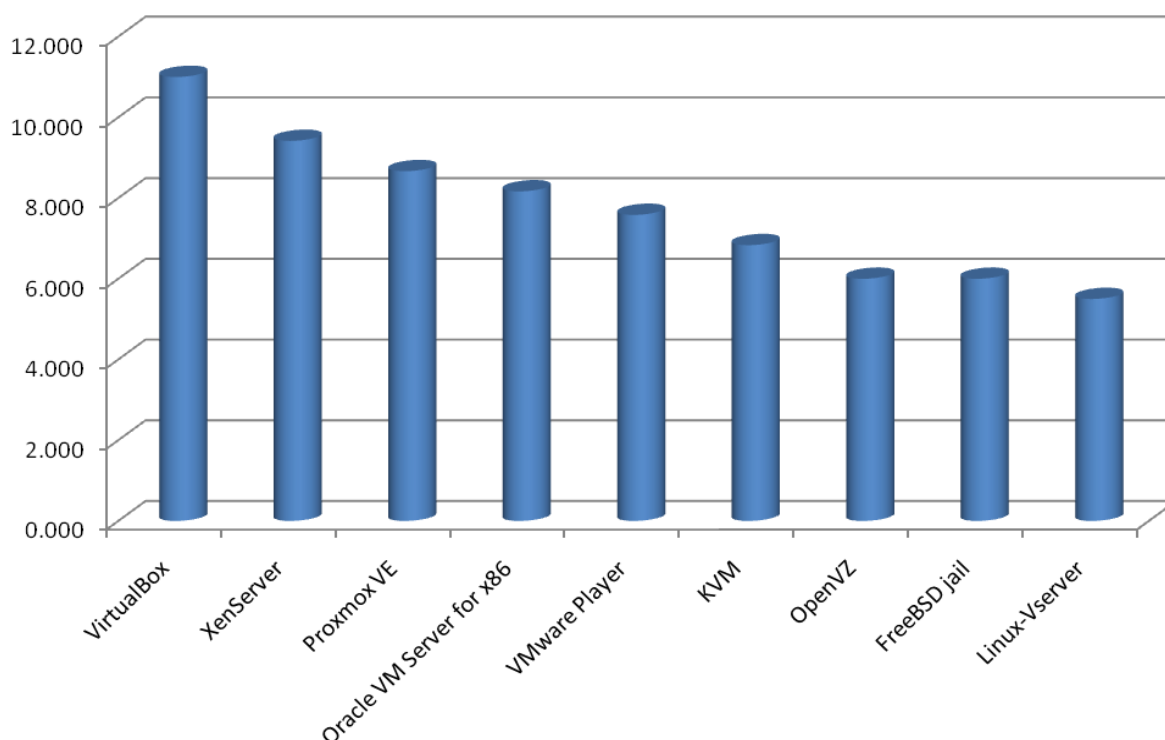


Рис. 2.8. Порівняльний аналіз програмних засобів віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики: графічне подання чисельних результатів оцінювання

За умови рівнозначності всіх показників, лідером за результатами порівняльного аналізу є Oracle VirtualBox (рис. 2.5). Втім, варто враховувати, що VirtualBox, будучи гіпервізором II типу, адаптований для віртуалізації настільних ОС, а не для віртуалізації серверів (хоч і має клієнтський веб-інтерфейс [71]), у зв'язку з чим у разі інтересу викладача і навчального закладу до організації сервера з віртуалізованими ОС є сенс також звернути увагу на гіпервізори I типу, гібридні гіпервізори та віртуальні контейнери.

Висновки до розділу 2

Описано загальну методику дослідження, розроблено модель застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики, уточнено критерії добору програмних засобів віртуалізації у навчанні бакалаврів інформатики Unix-подібних ОС та здійснено порівняльний аналіз програмних засобів віртуалізації на основі цих критеріїв.

Розроблена модель застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики складається з цільового, змістового, технологічного, діагностичного та аналітичного компонентів, послідовність яких циклічно повторюється з кожним проведенням курсу.

Цільовий компонент містить мету і завдання застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики, інформатичні компетентності бакалаврів інформатики щодо ОС. Складові цільового компоненту зумовлено відповідними галузевими стандартами вищої освіти України, проаналізованими дослідженнями з підготовки вчителів інформатики, світовим досвідом навчання ОС.

Змістовий компонент моделі відповідає змісту навчальної дисципліни «Операційні системи та системне програмування», сформованим, у свою чергу, згідно з інформатичними компетентностями щодо ОС, визначеними у межах цільового компоненту, й відповідними їм очікуваними результатами навчання. Компонент містить змістові модулі з навчальної дисципліни «Операційні системи та системне програмування», для яких є педагогічно виваженим застосування технологій віртуалізації.

Технологічним компонентом визначено форми, методи та засоби навчання, спрямовані на досягнення поставлених мети і завдань. Серед засобів навчання особливу увагу приділено засобам віртуалізації.

Діагностичний компонент передбачає діагностування рівня сформованості визначених компетентностей шляхом перевірки відповідних результатів навчання під час захисту лабораторних робіт, тестових опитувань з теорії ОС, тестового опитування «Unix-подібні операційні системи».

Діагностування доцільно здійснювати згідно з розробленими критеріями (теоретичний, адміністративний, програмістський, самонавчальний), показниками та рівнями (початковий, середній, достатній, високий) сформованості інформатичних компетентностей бакалаврів інформатики щодо ОС. Також у межах діагностичного компоненту передбачено виявлення й усунення технічних проблем, зокрема пов'язаних з використанням засобів віртуалізації, та профілактика виникнення цих проблем у майбутньому. Рекомендовано здійснювати документування успішних технічних рішень та запланованих змін до методики.

Аналітичний компонент містить аналіз ходу й результатів реалізації методики протягом поточної ітерації задля подальшого врахування підсумків аналізу на наступній ітерації і, як наслідок, поступового вдосконалення методики.

Добір програмних засобів віртуалізації у навчанні бакалаврів інформатики Unix-подібних ОС доцільно здійснювати на основі наступних критеріїв та показників: відповідність використовуваної технології віртуалізації потребам курсу (можливість формування інформатичних компетентностей щодо ОС; швидкість; надійність; універсальність щодо гостьових ОС; універсальність комп'ютерного класу; м'якість переходу для викладача та навчального закладу; уникнення надання студенту прав адміністратора; простота організації студентської самостійної роботи; оперативність переключення між ОС; простота вилучення/заміни ОС, що вивчається), універсальність (універсальність щодо основних ОС; універсальність щодо гостьових ОС; універсальність щодо технологій апаратної віртуалізації), ліцензійна чистота (вид ліцензії), функціональність інтерфейсу (наявність графічного інтерфейсу; можливість керування з командного рядка; наявність україномовного інтерфейсу; можливість віддаленого доступу), популярність та підтримка (середній пошуковий інтерес; засоби підтримки користувачів; дата випуску останнього стабільного релізу). На основі розроблених критеріїв проведено порівняльний аналіз ряду програмних засобів віртуалізації, а також

альтернативних варіантів навчання Unix-подібних ОС, котрі не передбачають безпосереднього застосування технологій віртуалізації. У разі, коли всі показники мають однакову вагу, найбільшу кількість позитивно проявлених показників отримав гіпервізор Oracle VirtualBox. Водночас, відзначено суттєву залежність одержаних результатів від вагових коефіцієнтів оцінюваних показників й зроблено висновок про потребу індивідуального добору засобів віртуалізації у кожному окремому випадку.

Основні результати проведеного у розділі 2 дослідження висвітлено у публікаціях: [40; 41; 42].

РОЗДІЛ 3.

МЕТОДИЧНІ ОСНОВИ ЗАСТОСУВАННЯ ТЕХНОЛОГІЙ ВІРТУАЛІЗАЦІЇ UNIX-ПОДІБНИХ ОПЕРАЦІЙНИХ СИСТЕМ У ПІДГОТОВЦІ БАКАЛАВРІВ ІНФОРМАТИКИ

3.1. Мета та завдання застосування технологій віртуалізації Unix-подібних операційних систем у підготовці бакалаврів інформатики

За [69, с. 295], цілепокладання розглядається як, «по-перше, прийняття і утримання цілей, поставлених іншою людиною перед суб'єктом, і, по-друге, – самостійна постановка цілей».

Мету і завдання методики застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики сформульовано відповідно до галузевого стандарту вищої освіти України за напрямом підготовки 6.040302 «Інформатика» [98; 99], проекту стандарту вищої освіти України для предметної спеціалізації 014.09 «Середня освіта (Інформатика)» для бакалаврського рівня освіти [100], проаналізованими дослідженнями з підготовки вчителів інформатики, світовим досвідом навчання ОС (керівними принципами розробки навчальних програм Computer Science Curricula 2013 [213], підручниками з ОС, відомостями з веб-ресурсів ЗВО). В основі мети та завдань застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики лежать інформатичні компетентності бакалаврів інформатики щодо ОС, визначені у п. 1.2.2.

Мета застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики полягає у створенні умов для формування в студентів інформатичних компетентностей щодо ОС.

Наведене формулювання мети узгоджується з гіпотезою дослідження: якщо навчання Unix-подібних ОС здійснювати на основі розробленої методики застосування технологій віртуалізації, то це дасть змогу підвищити результативність формування інформатичних компетентностей щодо ОС у бакалаврів інформатики.

Згідно з метою сформульовано *завдання, інформатичні компетентності бакалаврів інформатики щодо ОС*, які підлягають формуванню, та *результати навчання*.

Визначено чотири *завдання застосування Unix-подібних ОС у підготовці бакалаврів інформатики*.

- 1) Навчання основ теорії ОС.
- 2) Освоєння основ адміністрування ОС.
- 3) Навчання основ системного програмування.
- 4) Створення передумов для подальшої самоосвіти з ОС.

Кожному завданню відповідають *інформатичні компетентності бакалаврів інформатики щодо ОС*.

Завдання 1. Навчання основ теорії ОС.

Компетентності:

- знати та розуміти основні етапи, напрями та тенденції розвитку ОС;
- орієнтуватися у класифікації ОС, з урахуванням їх архітектури та сфери застосування;
- знати та розуміти теоретичні основи будови та функціонування ОС.

Завдання 2. Освоєння основ адміністрування ОС.

Компетентності:

- вміти використовувати інтерфейс користувача Unix-подібних ОС (графічний інтерфейс і/або інтерфейс командного рядка);
- вміти здійснювати налаштування мережного з'єднання у Unix-подібних ОС;
- вміти встановлювати програмне забезпечення у Unix-подібних ОС, виконувати його наступне оновлення та деінсталяцію;
- вміти відстежувати роботу процесів та потоків в ОС і керувати ними;
- вміти здійснювати моніторинг використання різних видів пам'яті в ОС;
- володіти основами роботи з файловою системою у Unix-подібних ОС;
- володіти основами роботи з налаштуваннями системи безпеки ОС.

Завдання 3. Навчання основ системного програмування.

Компетентності:

- володіти основами системного програмування мовою C++ у Linux (у Linux та Windows).

Завдання 4. Створення передумов для наступної самоосвіти з ОС.

Компетентності:

- вміти одержувати довідкові відомості щодо використання ОС та програмування у них.

Процес добору та структурування змісту, що відповідає вказаним вище завданням та компетентностям, розглянуто у п. 3.2.

3.2. Структура та зміст застосування технологій віртуалізації Unix-подібних операційних систем у підготовці бакалаврів інформатики

Зміст освіти – це «система наукових знань про природу, суспільство, людське мислення, практичних вмінь і навичок та способів діяльності, досвіду творчої діяльності, світоглядних, моральних, естетичних ідей та відповідної поведінки, якими людина повинна оволодіти шляхом навчання у навчальному закладі або самостійно» [175, с. 145]. Зміст є «конкретною відповіддю дидактики на питання "Що вчити?"» [101, с. 184].

Згідно із Законом України «Про вищу освіту», нормативний зміст підготовки здобувачів вищої освіти регламентується відповідним стандартом вищої освіти і формулюється у термінах результатів навчання [22, ст. 10 (3), п. 1]. Врахування вимог щодо змісту підготовки бакалаврів інформатики суттєво ускладнювалося відсутністю у період проведення дослідження затвердженого стандарту вищої освіти України, який би відповідав чинному законодавству. У зв'язку з цим, було враховано проект стандарту вищої освіти України для предметної спеціалізації 014.09 «Середня освіта (Інформатика)» для бакалаврського рівня освіти (2017 рік) [100], а також попередній галузевий стандарт для напряму підготовки 6.040302 «Інформатика» (2010 рік) [98; 99].

Оскільки вітчизняна нормативно-правова база щодо змісту підготовки бакалаврів інформатики на момент проведення дослідження перебувала у процесі формування, важливу роль у доборі змісту застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики відіграли керівні принципи розробки навчальних програм Computer Science Curricula 2013 (CS2013) [213] (п. 1.3). CS2013 містить, зокрема, перелік та опис змісту тематичних модулів і результатів навчання для підготовки за спеціальністю «Computer Science» [213, с. 135-141] (додаток Е). Зважаючи на спільність предметної галузі, вважаємо за доцільне часткове застосовування CS2013 під час розробки курсів і для бакалаврів інформатики, однак враховуючи відмінності між спеціальністю «Computer Science» та вітчизняною спеціальністю 014.09 «Середня освіта (Інформатика)» через наявність в останній також науково-педагогічної та професійно-педагогічної складових підготовки.

Зміст застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики тісно пов'язаний зі змістом навчальної дисципліни «Операційні системи та системне програмування», оскільки кожний елемент змісту дисципліни, який стосується Unix-подібних ОС, допускає пряме чи опосередковане використання технологій віртуалізації цих ОС, а кожний елемент змісту дисципліни загалом – використання технологій віртуалізації Unix-подібних чи інших ОС. Враховуючи ту обставину, що розгляд Unix-подібних ОС (наприклад, Linux) може здійснюватися шляхом порівняння їх з іншими ОС (наприклад, Windows), сформований зміст містить також елементи, пов'язані з ОС Windows.

Отже, зміст навчальної дисципліни «Операційні системи та системне програмування» для бакалаврів інформатики відбирався та структурувався відповідно до запропонованих у п. 3.1 мети і завдань застосування технологій віртуалізації Unix-подібних ОС, а також на основі Закону України «Про вищу освіту» [22], проекту стандарту вищої освіти України для предметної спеціалізації 014.09 «Середня освіта (Інформатика)» для бакалаврського рівня освіти

(2017 рік) [100], галузевого стандарту вищої освіти за напрямом підготовки «Інформатика» (2009 рік) [99], керівних принципів розробки навчальних програм Computer Science Curricula 2013 [213]. Досліджено та враховано досвід викладання курсу з ОС у провідних ЗВО світу, у тому числі Аерокосмічному університеті Ембрі-Ріддла, Гарвардському, Гельсінському, Єльському, Кембриджському університетах, Коледжі Вільямса, Колумбійському, Массачусетському, Принстонському, Рейк'явіцькому університетах, Технологічному інституті Джорджії, Стенфордському університеті, Університеті Арканзасу, Літл Рок, Університеті Каліфорнії, Берклі, Університеті Карнегі-Меллона, Федеральній вищій школі Цюриха, а також у системі масових онлайн курсів Udacity (див. п. 1.3) та низці закордонних та вітчизняних підручників з ОС, серед яких «Сучасні операційні системи» Е. Таненбаума та Г. Боса (Andrew S. Tanenbaum, Herbert Bos, 2012) [169], «Основи операційних систем» А. Сільбершаца, П. Б. Гелвіна і Г. Ганьє (Abraham Silberschatz, Peter B. Galvin, Greg Gagne, 2012) [275], «Операційні системи» Дж. Бекона і Т. Гарріса (Jean Bacon, Tim Harris, 2003) [201], «Внутрішня будова та принципи розробки операційних систем» В. Столлінгса (William Stallings, 2015) [278], «Операційні системи» В. А. Шеховцова (2008) [185], «Основи операційних систем» В. Є. Карпова, К. А. Конькова (2005) [75]. Було також враховано результати анкетування викладачів вітчизняних ЗВО щодо особливостей навчання Unix-подібних ОС бакалаврів інформатики та студентів інших спеціальностей, проведене на констатувальному етапі педагогічного експерименту (п. 4.2) та власний досвід викладання курсу з ОС бакалаврам інформатики у Житомирському державному університеті імені Івана Франка.

У підсумку було вибрано наступні змістові модулі.

Змістовий модуль 1. Початкові відомості про операційні системи.

Змістовий модуль 2. Процеси та потоки.

Змістовий модуль 3. Пам'ять.

Змістовий модуль 4. Файлові системи.

Змістовий модуль 5. Безпека.

Змістовий модуль 6. Планування.

Змістовий модуль 7. Міжпроцесова та міжпотоква взаємодія.

Змістовий модуль 8. Віртуалізація.

Запропонований перелік змістових модулів відповідає рекомендаціям з CS2013: з галузі знань «Операційні системи» (OS/Operating Systems) включено всі нормативні змістові модулі першого рівня (Core Tier-1), всі нормативні змістові модулі другого рівня (Core Tier-2), а також дві вибіркові змістові модулі (змістовий модуль 4 «Файлові системи», що відповідає однойменому вибіркового змістовому модулю з CS2013, і змістовий модуль 8 «Віртуалізація», яка відповідає вибіркового змістовому модулю з CS2013 «Віртуальні машини»). Нагадаємо, що автори CS2013 наполягають на розгляді всіх нормативних змістових модулів першого рівня (Core Tier-1), всіх чи майже всіх нормативних змістових модулів другого рівня (Core Tier-2) і частини вибірових змістових модулів (Elective). Відзначимо, що перелік питань, які підлягають вивченню у межах кожного змістового модуля, розподіл годин, зміст лабораторних робіт та очікувані результати навчання у низці аспектів відрізняються від рекомендацій з CS2013. Зокрема, серед вибраних питань відсутній детальний розгляд роботи пристроїв і написання простих драйверів і, навпаки, включено не зазначені у CS2013 питання та результати навчання, пов'язані з навичками користування ОС Linux (враховуючи відсутність досвіду роботи з Linux у значної частини цільової студентської аудиторії), що також відповідає меті і завданням застосування технологій віртуалізації Unix-подібних ОС, визначених у п. 3.1. Перелік питань адміністрування розширено: включено налаштування мережного з'єднання у Linux, редагування конфігураційних файлів тощо, тимчасом як у CS2013 безпосередньо йдеться лише про роботу з обліковими записами, групами і резервне копіювання. З іншого боку, меншу увагу, порівняно з CS2013, приділено основам системного програмування. Приміром, синхронізаційні примітиви розглядаються лише теоретично, не включено більшість питань з проектування ОС. Отже, в цілому зміст дисципліни «Операційні системи та системне програмування» для

бакалаврів інформатики добирався відповідно до вітчизняних умов та стандартів і водночас – з урахуванням західного досвіду навчання ОС.

У табл. К.1 (додаток К) наведено теми лекцій (у даному випадку вони відповідають назвам змістових модулів). Щодо кожної теми зазначено питання, які розглядаються у її межах; орієнтовна кількість відведених на її вивчення годин (за аналогією з CS2013, тут вказано *лекційні* години); очікувані результати навчання; ключові поняття, які формуються під час вивчення теми. Також у таблиці відображено відповідність між відібраними темами і змістовими модулями з галузі знань «Операційні системи», запропонованими у CS2013.

Табл. К.2 (додаток К) містить відомості про лабораторні роботи: тему лабораторної роботи; теми лекцій, які повинні передувати виконанню даної лабораторної роботи; засоби віртуалізації та інше програмне забезпечення, що використовується під час виконання роботи; очікувані результати навчання. Зазначимо, що згадані вище засоби віртуалізації й інше програмне забезпечення та їх зв'язок з лабораторними роботами буде детальніше розглянуто у п. 3.3 та п. 3.4.

Загалом пропонується чотирнадцять лабораторних робіт наступного змісту.

Лабораторна робота №1. Встановлення Linux. Історія виникнення ОС Linux. Поняття ядра Linux, дистрибутиву та випуску Linux. Історія створення дистрибутиву Ubuntu, його характерні особливості та основні версії. Процес встановлення Ubuntu за допомогою мережного інсталятора (без GUI).

Лабораторна робота №2. Знайомство з Linux. Огляд найтипівіших складових графічного інтерфейсу Linux на прикладі оболонки Xfce. Основні прийоми роботи з командним рядком Linux.

Лабораторна робота №3. Налаштування мережного з'єднання у Linux. Редагування найпростіших текстових файлів. Поняття конфігураційного файлу у Linux. Перегляд, аналіз та редагування конфігураційних файлів на прикладі налаштування мережного з'єднання.

Лабораторна робота №4. Встановлення ПЗ у Linux. Знайомство з пакунками Linux на прикладі системи пакунків у deb-сумісних дистрибутивах.

Базові операції з deb-пакунками засобами командного рядка та графічного інтерфейсу.

Лабораторна робота №5. Керування процесами у Linux.

Спостереження за роботою процесів та потоків у Linux. Керування процесами у Linux GUI-засобами та засобами командного рядка.

Лабораторна робота №6. Керування процесами у Windows.

Спостереження за роботою процесів та потоків у Windows. Керування процесами у Windows GUI-засобами та засобами командного рядка.

Лабораторна робота №7. Програмування процесів та потоків у Windows. Знайомство з інтерфейсом Win32 API, типами даних Windows та угорською нотацією. Створення процесів та потоків і програмування елементарних маніпуляцій з ними засобами Win32 API.

Лабораторна робота №8. Програмування процесів та потоків у Linux.

Створення процесів та потоків і виконання простих маніпуляцій з ними засобами системних викликів Linux.

Лабораторна робота №9. Моніторинг використання пам'яті та процесорного часу у Linux та Windows. Здійснення спостереження за використанням часу ЦП та пам'яті в ОС Linux та Windows.

Лабораторна робота №10. Робота з файлами у Linux. Особливості структури каталогів у Linux. Основні команди для роботи з файлами та каталогами. Перегляд та аналіз вмісту каталогу /dev, таблиці розділів (fdisk), відомостей про використання дискового простору (df). Операція монтування у Linux. Основи роботи з файловими архівами у Linux.

Лабораторна робота №11. Програмування роботи з файлами у Windows. Програмування основних маніпуляцій з файлами та каталогами засобами Win32 API.

Лабораторна робота №12. Програмування роботи з файлами у Linux. Програмування основних маніпуляцій з файлами та каталогами засобами системних викликів Linux.

Лабораторна робота №13. Користувачі та групи у Linux. Робота з обліковими записами та групами користувачів в Ubuntu засобами командного рядка та графічного інтерфейсу. Файлові атрибути в Unix-подібних ОС, їх інтерпретація та зміна. Рядок повноважень.

Лабораторна робота №14. Користувачі та групи у Windows. Робота з обліковими записами та групами користувачів у Windows. Списки керування доступом у Windows.

Зазначені у табл. К.1 та табл. К.2 результати навчання обумовлюють вимоги до захисту лабораторних робіт (переліки вмінь, контрольні запитання, програмні коди та інші файли, які має надати студент під час захисту роботи), запитання контрольних тестових опитувань за матеріалами лекцій, теоретичні запитання та практичні завдання в екзаменаційних білетах. З іншого боку, вказані результати навчання деталізують визначені у п. 3.1 мету й завдання застосування технологій віртуалізації Unix-подібних ОС та компетентності, що відповідають кожному завданню. Повний опис утвореної у такий спосіб чотирирівневої структури (мета – завдання – компетентності – результати навчання) наведено у додатку Л.

Структурований зміст застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики лежить в основі навчальної та робочої програм дисципліни «Операційні системи та системне програмування» для студентів напряму підготовки 6.040302 «Інформатика*», нині – спеціальності 014.09 «Середня освіта (Інформатика)», Житомирського державного університету імені Івана Франка, з якою можна ознайомитись у додатках Г.

Важливо зауважити, що у даному випадку змістом навчальної дисципліни «Операційні системи та системне програмування» передбачено поряд із вивченням ОС Linux вивчення також і ОС Windows, внаслідок чого відповідні результати навчання включено до загального переліку очікуваних результатів навчання. Результати навчання, пов'язані з практичною роботою в ОС Windows, можуть бути відсутні у цьому переліку, якщо окремо взятим курсом з ОС таку

роботу не передбачено. Водночас, пропоновану методику розроблено з урахуванням того, що ОС, визначені навчальною програмою для практичного освоєння, не обов'язково лишаються незмінними. Те саме стосується обраних змістових модулів, тем лабораторних робіт, детального переліку питань тощо. Ці та інші особливості конкретного курсу з ОС можуть варіювати у різних ЗВО або й в одному ЗВО у різні навчальні роки. Ця обставина є одним з факторів вироблення варіативного підходу, описаного у п. 3.3.

Отже, запропонований зміст дисципліни «Операційні системи та системне програмування» відображає, з одного боку, чільне місце Unix-подібних ОС та технологій віртуалізації у навчанні курсу з ОС, й з іншого – тісний зв'язок Unix-подібних ОС з іншими ОС і технологіями віртуалізації.

Діагностування рівнів сформованості компетентностей здійснюється відповідно до *критеріїв та показників сформованості інформатичних компетентностей бакалаврів інформатики щодо ОС*, поданих у таблиці 3.1 й опублікованих також у роботі [160].

Оцінювання відбувається за чотирма *рівнями сформованості інформатичних компетентностей бакалаврів інформатики щодо ОС*, загальні характеристики яких є такими.

Початковим рівнем («2») передбачено наявність фрагментарних знань і мінімальну самостійність у виконанні практичних завдань.

Середній рівень («3») свідчить про володіння необхідним мінімумом знань та деяку самостійність у виконанні практичних завдань.

Достатньому рівню («4») відповідає впевнене володіння основним обсягом знань та помірна самостійність у виконанні практичних завдань.

Високим рівнем («5») передбачено вільне володіння визначеним програмою обсягом знань і суттєва самостійність у виконанні практичних завдань, а також схильність до аналітичного й творчого опрацювання матеріалу, пошуку власних способів виконання завдань.

Детальні характеристики перелічених рівнів наведено у таблиці 3.2.

Таблиця 3.1

**Критерії та показники сформованості інформатичних компетентностей
бакалаврів інформатики щодо ОС**

Критерії	Показники
1. Теоретичний <i>Знання основ теорії ОС</i>	1.1. Знання та розуміння основних етапів, напрямів та тенденцій розвитку ОС 1.2. Орієнтування у класифікації ОС, з урахуванням їх архітектури та сфери застосування 1.3. Знання та розуміння теоретичних основ будови та функціонування ОС
2. Адміністративний <i>Володіння основами адміністрування ОС</i>	2.1. Вміння використовувати інтерфейс користувача Unix-подібних ОС 2.2. Вміння здійснювати налаштування мережного з'єднання у Unix-подібних ОС 2.3. Вміння встановлювати програмне забезпечення у Unix-подібних ОС, виконувати його наступне оновлення та деінсталяцію 2.4. Вміння відстежувати роботу процесів та потоків в ОС і керувати ними 2.5. Вміння здійснювати моніторинг використання різних видів пам'яті в ОС 2.6. Володіння основами роботи з файловою системою у Unix-подібних ОС 2.7. Володіння основами роботи з налаштуваннями системи безпеки ОС
3. Програмістський <i>Володіння основами системного програмування</i>	3.1. Володіння основами системного програмування в ОС Linux мовою C++ із застосуванням системних бібліотек 3.2. Володіння основами системного програмування в ОС Windows мовою C++ із застосуванням API-інтерфейсу.
4. Самонавчальний <i>Здатність до наступної самоосвіти з ОС</i>	4.1. Вміння одержувати довідкові відомості щодо використання ОС та програмування у них вбудованими засобами ОС 4.2. Вміння одержувати довідкові відомості щодо використання ОС та програмування у них зі спеціалізованих веб-ресурсів

Таблиця 3.2

**Рівні сформованості інформатичних компетентностей
бакалаврів інформатики щодо ОС**

Критерії / Показники	Параметри (Рівні)			
	«2» (початковий рівень)	«3» (середній рівень)	«4» (достатній рівень)	«5» (високий рівень)
1. Теоретичний <i>Знання основ теорії ОС</i>				
1.1. Знання та розуміння основних етапів, напрямів та тенденцій розвитку ОС	Має фрагментарні уявлення про основні етапи, напрями та тенденції розвитку ОС	Має базові уявлення про основні етапи, напрями та тенденції розвитку ОС	Добре орієнтується в основних етапах, напрямках та тенденціях розвитку ОС	Здатний аналізувати основні етапи, напрями та тенденції розвитку ОС. Схильний самостійно виокремлювати тенденції розвитку ОС та робити прогнози на майбутнє
1.2. Орієнтування у класифікації ОС, з урахуванням їх архітектури та сфери застосування	Має фрагментарні уявлення про класифікацію ОС	Розрізняє основні класи та архітектури ОС та може пояснити деякі характерні відмінності між ними	Впевнено розрізняє класи та архітектури ОС, аргументовано визначає, до якого класу (класів) належить конкретна ОС	Здатний порівнювати системи різних класів та архітектур. Усвідомлює умовність класифікації й наводить приклади ОС, що можуть бути віднесені до кількох класів
1.3. Знання та розуміння теоретичних основ будови та функціонування ОС	Має фрагментарні уявлення про теоретичні основи будови та функціонування ОС. Називає деякі складові ОС та їх призначення	Має базові уявлення про теоретичні основи будови та функціонування ОС. Знає більшість основних складових ОС та їх призначення	Впевнено знає теоретичні основи будови та функціонування ОС. Може пояснити, як працюють основні складові ОС	Вільно орієнтується в теоретичних основах будови та функціонування ОС. Здатний навести приклади конкретних реалізацій

Критерії / Показники	Параметри (Рівні)			
	«2» (початковий рівень)	«3» (середній рівень)	«4» (достатній рівень)	«5» (високий рівень)
2. Адміністративний <i>Володіння основами адміністрування ОС</i>				
2.1. Вміння використовувати інтерфейс користувача Unix-подібних ОС (графічний інтерфейс, інтерфейс командного рядка)	З активною допомогою викладача використовує інтерфейс користувача (* зі значним переважанням GUI)	З помірною допомогою викладача використовує інтерфейс користувача (* з базовим володінням і GUI, і CLI, можливо, із наданням переваги GUI)	З епізодичною допомогою викладача та самостійним залученням довідкових матеріалів використовує інтерфейс користувача (* з добрим володінням і GUI, і CLI, можливо, із наданням переваги GUI)	Вільно чи з епізодичним самостійним залученням довідкових матеріалів використовує інтерфейс користувача (* з вільним володінням і GUI, і CLI, можливо, із наданням переваги CLI)
2.2. Вміння здійснювати налаштування мережного з'єднання у Unix-подібних ОС	З активною допомогою викладача переглядає поточні налаштування мережного з'єднання. Пояснює окремі готові налаштування (* зі значним переважанням GUI)	Пояснює основні готові налаштування мережного з'єднання, у тому числі загалом орієнтується у вмісті відповідних конфігураційних файлів. З помірною допомогою викладача налаштовує з'єднання, що потребують мінімальних налаштувань, наприклад, DHCP-з'єднання (* з базовим володінням і GUI, і CLI, можливо, із наданням переваги GUI)	Пояснює більшість готових налаштувань мережного з'єднання, у тому числі добре орієнтується у вмісті відповідних конфігураційних файлів. З епізодичною допомогою викладача та самостійним залученням довідкових матеріалів налаштовує більшість вивчених типів з'єднання (* з добрим володінням і GUI, і CLI, можливо, із наданням переваги GUI)	Пояснює будь-які (у межах навчальної програми) готові налаштування мережного з'єднання, у тому числі вільно орієнтується у вмісті відповідних конфігураційних файлів. Вільно чи з епізодичним самостійним залученням довідкових матеріалів налаштовує всі вивчені типи з'єднання (* з вільним володінням і GUI, і CLI, можливо, із наданням переваги CLI)

Критерії / Показники	Параметри (Рівні)			
	«2» (початковий рівень)	«3» (середній рівень)	«4» (достатній рівень)	«5» (високий рівень)
2.3. Вміння встановлювати програмне забезпечення у Unix-подібних ОС, виконувати його наступне оновлення та деінсталяцію	З активною допомогою викладача визначає, яке програмне забезпечення вже встановлене у системі, та виконує окремі кроки щодо встановлення та деінсталяції нового програмного забезпечення (* зі значним переважанням GUI)	З помірною допомогою викладача встановлює нове програмне забезпечення за заздалегідь відомими назвами пакунків, найпростіші варіанти його деінсталяції та оновлення. Розуміє частину налаштувань системи керування пакунками (* з базовим володінням і GUI, і CLI, можливо, із наданням переваги GUI)	З епізодичною допомогою викладача та залученням довідкових матеріалів встановлює нове ПЗ, здійснює пошук потрібних пакунків, їх оновлення та різні варіанти їх деінсталяції. Розуміє більшість налаштувань системи керування пакунками, з деякою допомогою викладача їх змінює (* з добрим володінням і GUI, і CLI, можливо, із наданням переваги GUI)	Вільно чи з епізодичним залученням довідкових матеріалів встановлює нове ПЗ, здійснює пошук пакунків, їх оновлення та різні варіанти їх деінсталяції. Розуміє всі передбачені навчальною програмою налаштування системи керування пакунками, самостійно чи з залученням довідкових матеріалів змінює їх (* з вільним володінням і GUI, і CLI, можливо, із наданням переваги CLI)
2.4. Вміння відстежувати роботу процесів та потоків в ОС і керувати ними	З активною допомогою викладача переглядає список запущених процесів та може виконувати окремі найпростіші операції з ними (* зі значним переважанням GUI)	З помірною допомогою викладача переглядає список запущених процесів, знає призначення деяких стандартних процесів, виконує основні операції з процесами (* з базовим володінням і GUI, і CLI, можливо, із наданням переваги GUI)	З епізодичною допомогою викладача та залученням довідкових матеріалів виводить та фільтрує список запущених процесів, знаходить процес за чіткими відомостями про нього, виконує більшість передбачених навчальною програмою операцій з процесами (* з добрим володінням і GUI, і CLI, можливо, із наданням переваги GUI)	Вільно чи з епізодичним залученням довідкових матеріалів аналізує список запущених процесів, знаходить процес за уривчастими відомостями про нього, виконує всі передбачені програмою операції з процесами (* з вільним володінням і GUI, і CLI, можливо, із наданням переваги CLI)

<i>Критерії / Показники</i>	<i>Параметри (Рівні)</i>			
	<i>«2» (початковий рівень)</i>	<i>«3» (середній рівень)</i>	<i>«4» (достатній рівень)</i>	<i>«5» (високий рівень)</i>
2.5. Вміння здійснювати моніторинг використання різних видів пам'яті в ОС	З активною допомогою викладача переглядає відомості про використання пам'яті, пояснює деякі фрагменти виводу (* зі значним переважанням GUI)	З помірною допомогою викладача переглядає відомості про використання пам'яті, пояснює основні фрагменти виводу (* з базовим володінням і GUI, і CLI, можливо, із наданням переваги GUI)	З епізодичною допомогою викладача та залученням довідкових матеріалів переглядає відомості про використання пам'яті, пояснює основні передбачені навчальною програмою фрагменти виводу (* з добрим володінням і GUI, і CLI, можливо, із наданням переваги GUI)	Вільно чи з епізодичним залученням довідкових матеріалів переглядає відомості про використання пам'яті, пояснює всі передбачені програмою фрагменти виводу. Робить висновки щодо розподілу пам'яті у даній системі (* з вільним володінням і GUI, і CLI, можливо, із наданням переваги CLI)
2.6. Володіння основами роботи з файловою системою у Unix-подібних ОС	З активною допомогою викладача здійснює навігацію файловою системою, виконує окремі операції з файлами (* зі значним переважанням GUI)	З помірною допомогою викладача здійснює навігацію файловою системою, виконує основні операції з файлами (* з базовим володінням і GUI, і CLI, можливо, із наданням переваги GUI)	З епізодичною допомогою викладача та залученням довідкових матеріалів здійснює навігацію файловою системою, виконує більшість передбачених навчальною програмною операцій з файлами (* з добрим володінням і GUI, і CLI, можливо, із наданням переваги GUI)	Вільно чи з епізодичним залученням довідкових матеріалів здійснює навігацію файловою системою, виконує всі передбачені навчальною програмною операцій з файлами (* з вільним володінням і GUI, і CLI, можливо, із наданням переваги CLI)

Критерії / Показники	Параметри (Рівні)			
	«2» (початковий рівень)	«3» (середній рівень)	«4» (достатній рівень)	«5» (високий рівень)
2.7. Володіння основами роботи з налаштуваннями системи безпеки ОС	З активною допомогою викладача переглядає налаштування облікових записів, груп, розмежування доступу для файлів та пояснює деякі з них (* зі значним переважанням GUI)	З помірною допомогою викладача переглядає налаштування облікових записів, груп, розмежування доступу для файлів, пояснює базові налаштування та здатний вносити в ці налаштування нескладні зміни (* з базовим володінням і GUI, і CLI, можливо, із наданням переваги GUI)	З епізодичною допомогою викладача та залученням довідкових матеріалів переглядає налаштування облікових записів, груп, розмежування доступу для файлів, пояснює більшість передбачених навчальною програмою налаштувань та здатний вносити у них зміни, які потребують основних уявлень про політики безпеки (* з добрим володінням і GUI, і CLI, можливо, із наданням переваги GUI)	Вільно чи з епізодичним залученням довідкових матеріалів переглядає налаштування облікових записів, груп, розмежування доступу для файлів, пояснює всі передбачені навчальною програмою налаштування та здатний вносити у них зміни, що потребують добре розуміння політик безпеки (* з вільним володінням і GUI, і CLI, можливо, із наданням переваги CLI)
3. Програмістський <i>Володіння основами системного програмування</i>				
3.1. Володіння основами системного програмування в ОС Linux мовою C++ із застосуванням системних бібліотек	Активно використовуючи довідкові матеріали, може пояснити окремі фрагменти готового програмного коду. Знає призначення окремих бібліотечних функцій, що відповідають темі лабораторної роботи. Здатний виправити деякі помилки у коді, виявлені під час захисту, з активною допомогою викладача	Активно використовуючи довідкові матеріали, здатний написати програму для реалізації завдань, близьких до наданих прикладів. Впевнено орієнтується в окремих фрагментах коду програми. З активною допомогою викладача виправляє більшість помилок у коді	Здатний самостійно чи з епізодичним використанням довідкових матеріалів написати програму для реалізації завдань, близьких до наданих прикладів. Впевнено орієнтується у більшості коду програми. Виправляє всі чи майже всі помилки у коді.	Здатний самостійно чи з епізодичним використанням довідкових матеріалів написати програму для реалізації завдань, суттєво відмінних від наданих прикладів. Впевнено орієнтується в усьому коді програми. Самостійно виправляє всі помилки у коді.
3.2. Володіння основами системного програмування в ОС Windows мовою C++ із застосуванням API.				

Критерії / Показники	Параметри (Рівні)			
	«2» (початковий рівень)	«3» (середній рівень)	«4» (достатній рівень)	«5» (високий рівень)
4. Самонавчальний <i>Здатність до наступної самоосвіти з ОС</i>				
4.1. Вміння одержувати довідкові відомості щодо використання ОС та програмування у них вбудованими засобами ОС	Вміє викликати принаймні одну команду одержання допомоги. Знає назву щонайменше одного релевантного довідкового ресурсу	Вміє викликати команди одержання допомоги та загалом орієнтується у відомостях, що виводяться цими командами. Здатний продуктивно залучати довідкові ресурси до вирішення простих проблем (наприклад, з'ясувати синтаксис бібліотечної функції)	Вміє викликати команди одержання допомоги та добре орієнтується у відомостях, що виводяться цими командами. Здатний продуктивно залучати довідкові ресурси до вирішення проблем помірної складності (наприклад, вивчити досвід інших користувачів, котрі вирішували таку саму або подібну проблему)	Вміє викликати команди одержання допомоги та вільно орієнтується у відомостях, що виводяться цими командами. Здатний продуктивно залучати довідкові ресурси до вирішення складних та нестандартних проблем (наприклад, на основі досвіду користувачів, котрі вирішували дещо подібні проблеми, у тому числі не успішно, виробити власне успішне чи частково успішне рішення)
4.2. Вміння одержувати довідкові відомості щодо використання ОС та програмування у них зі спеціалізованих веб-ресурсів				

Надамо деякі коментарі та пояснення щодо відомостей, поданих у таблиці 3.2.

Коментарі зі знаком *, подані у дужках, стосуються випадків, коли програма курсу включає практичне використання графічного інтерфейсу користувача (GUI) в Unix-подібній ОС. Якщо застосування GUI не передбачене дисципліною загалом чи у межах окремих лабораторних робіт, означені коментарі в дужках до уваги не беруться.

Під *довідковими матеріалами* найчастіше маються на увазі спеціальні довідкові команди (man, help) й офіційні та неофіційні видання і ресурси, присвячені синтаксису та особливостям використання команд, бібліотечних функцій, об'єктів, конфігураційних файлів тощо, і котрі також можуть містити приклади роботи з цими командами, функціями, об'єктами, файлами. Описувана методика передбачає надання студентам права використовувати описані вище засоби під час виконання ними всіх форм роботи, у тому числі й форм контролю (передусім практичних завдань і з обов'язковим урахуванням ступеню самостійності під час оцінювання, що відображено у таблиці). Також до категорії довідкових матеріалів може бути зараховано й публікації, що описують досвід вирішення окремих проблем (користувацьких, адміністративних, програмістських), зокрема спеціалізовані мережні спільноти (Stack Overflow, Stack Exchange тощо) і які нерідко містять послідовності команд і фрагменти готового програмного коду. Заохочується важливість фіксування та зазначення джерела, з якого було одержано відомості. Видання та ресурси, присвячені теоретичним основам будови та функціонування ОС, також можуть бути класифіковані як довідкові. Їх використання поза заходами контролю описуваною методикою заохочуються, натомість у межах цих заходів – здебільшого не дозволяється.

Запропоновані критерії не передбачають сумування на їх основі єдиного балу. Вони призначені для оцінювання окремих видів роботи (лабораторні роботи, тестування, екзамен тощо). Враховуючи, що запропонована методика передбачає досить високу гнучкість щодо змісту навчальної програми, під час

оцінювання можуть бути застосовані не всі наведені у таблиці критерії/показники, або, навпаки, може здійснюватися більша деталізація наявних критеріїв та показників. У разі доповнення даної системи критеріїв загальні характеристики її рівнів зберігаються.

3.3. Варіативний підхід до застосування засобів віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики

У вступі до даної роботи було відзначено проблему індивідуального добору засобів віртуалізації для навчання окремих дисциплін, у тому числі для навчання ОС бакалаврів інформатики, а також недостатня розробленість відповідного навчально-методичного забезпечення, зокрема рекомендацій стосовно добору віртуалізаційного ПЗ, нестачу інформаційно-дидактичних матеріалів для ознайомлення викладачів та студентів з технологіями віртуалізації.

У п. 2.3 було запропоновано та обґрунтовано критерії добору програмних засобів віртуалізації у навчанні бакалаврів інформатики Unix-подібних ОС. На основі сформульованих критеріїв здійснено порівняльний аналіз низки програмних засобів віртуалізації на базі різних технологій віртуалізації. Результати порівняльного аналізу виявили суттєву залежність результатів від вагових коефіцієнтів оцінюваних показників, які, у свою чергу, можуть суттєво відрізнятися у кожному окремому випадку. Іншими словами, програмні засоби віртуалізації, котрі найкраще підходять у деякому випадку, залежать від того, які показники є впливовішими, а які менш важливими.

Помітні розбіжності в оцінюванні важливості запропонованих показників різними викладачами було зафіксовано у ході анкетування викладачів ЗВО України, які викладали дисципліну «Операційні системи та системне програмування» та інші подібні дисципліни майбутнім бакалаврам інформатики, а також викладачів, які викладали аналогічні дисципліни студентам інших напрямів підготовки / спеціальностей (результати анкетування детально розглянуто та проаналізовано у п. 4.2). Було виявлено низку суттєвих

відмінностей у навчанні ОС у ЗВО України. Особливо суттєво відрізнялися: ОС, що вивчалися; категорії використовуваного віртуалізаційного ПЗ та конкретні застосовувані засоби віртуалізації; досвід застосування викладачами засобів віртуалізації; технічні умови, зокрема доступні викладачам альтернативи засобам віртуалізації та наявність процесорів із підтримкою апаратної віртуалізації; співвідношення застосування під час лабораторних робіт графічного інтерфейсу користувача та інтерфейсу командного рядка; оцінювання викладачами низки факторів добору засобів віртуалізації.

Під час бесід зі студентами, які вивчають курс з ОС, і викладачами курсу з ОС (п. 4.2), а також на основі власного досвіду викладання курсу з ОС було відзначено неоднаковість умов, у яких здійснюється самостійна робота студентів, проблеми частих апаратних і програмних збоїв, пов'язаних із застосуванням засобів віртуалізації. Вказані обставини підштовхують до використання альтернативних засобів віртуалізації поряд з основними, причому такий прийом знаходить підтвердження у закордонному досвіді викладання ОС (п. 1.3).

Застосування альтернативних засобів віртуалізації та індивідуальний добір засобів віртуалізації для курсу з ОС покладені в основу *варіативного підходу до застосування засобів віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики*.

Варіативний підхід до застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики полягає у поєднанні кількох засобів віртуалізації та, можливо, варіантів навчання ОС без використання віртуалізації задля відповідності індивідуальним особливостям проведення курсу й індивідуальним потребам студентів і забезпечення стійкості у разі програмних та апаратних збоїв.

Підхід передбачає наступні **завдання**.

- 1) Ознайомити викладачів та студентів з різноманітністю засобів віртуалізації (відмінності, переваги, обмеження).

- 2) Навести приклад поєднання кількох засобів віртуалізації у межах курсу з ОС.
- 3) Здійснити адаптування навчально-методичних матеріалів до використання різних засобів віртуалізації.

Комбінація засобів віртуалізації є реалізацією завдання 2 (навести приклад поєднання кількох засобів віртуалізації у межах одного курсу з ОС). На рис. 3.1 подано приклад комбінації засобів віртуалізації та варіантів навчання ОС, які не передбачають віртуалізації. Запропонована комбінація використовувалась у ході формувального етапу педагогічного експерименту під час навчання студентів експериментальної групи (п. 4.3).

	Linux	Windows
Адміністрування	VirtualBox Amazon EC2 ☒ без віртуалізації	Amazon EC2 ☒ без віртуалізації
Програмування	VirtualBox ± Code::Blocks Amazon EC2 ☒ без віртуалізації	☒ без віртуалізації + Code::Blocks Amazon EC2 + Code::Blocks

Рис. 3.1. Комбінація засобів віртуалізації для навчання ОС бакалаврів інформатики

Комбінація передбачає вивчення двох ОС – Linux та Windows, а також демонструє залучення засобів віртуалізації для реалізації завдань (2) та (3) запропонованої методики: (2) *освоєння основ адміністрування ОС*; (3) *навчання основ системного програмування* (рядки «Адміністрування» та «Програмування» відповідно). Зауважимо, що завдання (1) *навчання основ теорії ОС* досягається передусім у межах лекційної складової курсу, а завдання (4) *створення передумов для наступної самоосвіти з ОС* – у межах усього курсу, у тому числі й під час виконання лабораторних робіт, що стосуються

адміністрування ОС, та робіт, присвячених основам системного програмування для ОС.

За потреби (зокрема, у разі зменшення обсягу годин) комбінацію може бути модифіковано шляхом видалення з неї одного і більше прямокутників. Наприклад, комбінація допускає розгляд лише ОС Linux (рис. 3.2) або лише основ адміністрування (рис. 3.3).

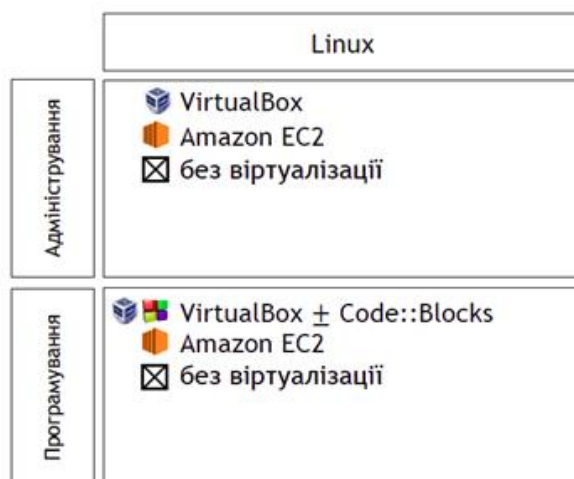


Рис. 3.2. Комбінація засобів віртуалізації для навчання ОС бакалаврів інформатики: модифікація для випадків, коли лабораторні роботи присвячено лише ОС Linux



Рис. 3.3. Комбінація засобів віртуалізації для навчання ОС бакалаврів інформатики: модифікація для випадків, коли лабораторні роботи присвячено лише основам адміністрування ОС

У кожному прямокутнику вказано засоби віртуалізації. Описуваною комбінацією пропонується використання двох різних засобів віртуалізації.

1) Oracle VirtualBox. Належить до гіпервізорів II типу. Встановлюється локально, на тих комп'ютерах, за якими працюють студенти, у тому числі, за

потреби, на їхніх власних ноутбуках чи домашніх комп'ютерах. Основна ОС, вже встановлена на комп'ютерах, у такому разі лишається без змін.

2) Amazon EC2. Хмарний сервіс моделі IaaS (Infrastructure as a Service – інфраструктура як сервіс), що надає можливість працювати з віртуальними машинами («інстансами», за термінологією Amazon) у хмарі. Сервіс має рівень безкоштовного користування (Free Tier), доступний протягом року з дня реєстрації.

Засоби віртуалізації використовуються у поєднанні з інтегрованим середовищем розробки Code::Blocks. Крім того, у кожному з прямокутників наявний варіант без віртуалізації (наприклад, якщо вивчається Linux, і студент встановлює Linux на власний комп'ютер як другу ОС поряд із Windows).

Верхній засіб у кожному прямокутнику є **основним**, решта – **альтернативними**. Основний засіб застосовується у більшості ситуацій. Альтернативні засоби можуть бути використані в окремих випадках, приміром, у разі виникнення технічних проблем з основним засобом (збій, неможливість працювати у звичному комп'ютерному класі, проблеми зі встановленням віртуалізаційного ПЗ на студентський ноутбук тощо).

Розглянута комбінація засобів віртуалізації може використовуватись у незміненому вигляді, бути модифікованою відповідно до наявних потреб, а також виступати у ролі ілюстрації варіанту поєднання засобів віртуалізації та варіантів навчання ОС без віртуалізації у межах одного курсу.

Розглянемо реалізацію *завдання 3* пропонованого варіативного підходу (адаптація навчально-методичних матеріалів до використання кількох засобів віртуалізації). Успішне поєднання кількох засобів віртуалізації потребує наявності відповідного навчально-методичного забезпечення, передусім інструктивно-методичних матеріалів до лабораторних робіт, які б, з одного боку, не були б прив'язані до єдиного засобу віртуалізації, а з іншого – містили інструкції з використання щонайменше двох різних засобів віртуалізації, адаптовані до потреб курсу.

Означену потребу було враховано під час написання методичного посібника «Операційні системи та системне програмування» (Житомир, «Рута», 2016) [44].

У процесі добору змісту методичного посібника основними орієнтирами стали:

- орієнтування на мету, завдання, компетентності та результати навчання методики застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики (п. 3.1-3.2);
- включення як питань, пов'язаних з Linux, так і питань, пов'язаних з Windows (з пріоритетом Linux як Unix-подібній ОС);
- поєднання засобів командного рядка та засобів графічного інтерфейсу (із пріоритетом засобам командного рядка, враховуючи їх відносну сталість, стандартизованість і невимогливість до системних ресурсів, порівняно із засобами графічного інтерфейсу);
- абстрагування від конкретних версій/дистрибутивів/випусків ОС та інтерфейсів (там, де це можливо), орієнтування на стандарти (зокрема POSIX), налаштування через конфігураційні файли тощо;
- абстрагування від засобів навчання (гіпервізори I типу, гіпервізори II типу, гібридні гіпервізори, віртуальні контейнери, хмарні сервіси, середовища програмування тощо) із одночасним наданням базових інструкцій щодо роботи з кількома різними засобами.

Посібник поділено на дві основні частини – «Інструментарій» та «Лабораторні роботи». Частина «Інструментарій» містить *самостійні роботи*, які описують основи роботи з засобами віртуалізації (VirtualBox, Amazon EC2) та інтегрованим середовищем розробки Code::Blocks.

Самостійні роботи можуть:

- а) виконуватися обов'язково і підлягати захисту;
- б) виконуватися обов'язково, але не підлягати захисту (знання, уміння та навички, здобуті під час опрацювання самостійних робіт, проявляються під час застосування відповідного засобу для виконання лабораторних робіт);

в) використовуватися як самовчитель і застосовуватися лише за потреби (наприклад, коли засіб, використовуваний раніше, з тих чи інших причин недоступний, і потрібна альтернатива);

г) не використовуватися взагалі, якщо відповідний засіб не застосовується.

Виконання самостійних робіт може відбуватися:

- у межах окремо відведених аудиторних годин;
- у межах годин, відведених на самостійну роботу студентів;
- не відбуватися взагалі, якщо відповідний засіб не використовується.

Методичний посібник побудовано у такий спосіб, аби викладач мав змогу відбирати самостійні і лабораторні роботи, зважаючи на навчальні плани та навчальні програми, доступні засоби навчання, рівень підготовки студентів, власні погляди тощо.

Частина «Лабораторні роботи» містить лабораторні роботи, складені відповідно до наступних *принципів*.

Зв'язок між лабораторними роботами та програмними засобами.

Інструкції до лабораторних робіт містять зауваження щодо засобів віртуалізації та інших програмних засобів, які використовуються (можуть бути використані) під час виконання кожної роботи (рис. 3.4).

Інструменти:

- а)** (графічний інтерфейс) VirtualBox + віртуальна машина зі встановленою Xubuntu 14.04 LTS + (рекомендовано) Інтернет-з'єднання;
- б)** (лише командний рядок) VirtualBox + віртуальна машина зі встановленою Ubuntu 14.04 LTS без GUI + (рекомендовано) Інтернет-з'єднання;
- в)** (лише командний рядок) Amazon EC2 + віртуальна машина EC2 t2.micro зі встановленою Ubuntu Server 14.04 LTS + (обов'язково) Інтернет-з'єднання.

Рис. 3.4. Методичний посібник «Операційні системи та системне програмування» для бакалаврів інформатики: рекомендації щодо ПЗ на початку лабораторної роботи

Розмежованість GUI-засобів та засобів командного рядка. В інструкціях до лабораторних робіт використовується спеціальне маркування блоків, які стосуються графічного інтерфейсу, та блоків, пов'язаних з

командним рядком (рис. 3.5). Це дозволяє пропускати блоки певного типу, якщо їх опрацювання не передбачено.

Виконати програму клацніть по знаку . З'явиться список встановлених програм, серед яких виберіть потрібну.



Завдання №2.2.17.

1) Перегляньте відкритий у завданні №2.2.16 файл /etc/sudoers. Відшукайте там одного користувача та дві групи, яким за замовчуванням дозволяється набувати прав адміністратора через механізм sudo.

Дуже важливо! Будьте обережні. На живій машині цей файл можна редагувати лише у виняткових випадках і лише тоді, коли ви точно знаєте, що робите. Якщо у майбутньому вам треба буде надати право використовувати sudo користувачу, такого права позбавленому, є інші, безпечніші способи це зробити²².

2) Закрийте програму Mousepad. Якщо ви випадково внесли до файлу /etc/sudoers якісь зміни, і вам пропонується зберегти їх, – **не зберігайте**.



Завдання №2.2.18.

1) Використовуючи вікно Пошук встановлених програм, запустіть програму Mousepad без прав адміністратора і без файлу.

2) Закрийте вікно програми Mousepad.



Набуття прав адміністратора у командному рядку

Для виконання тієї чи іншої команди з правами адміністратора допишіть перед нею sudo:

sudo команда, для виконання якої потрібні права адміністратора параметри

Система запитає пароль – введіть його (пам'ятайте, що у командному рядку паролі під час введення не відображаються взагалі ніяк – навіть зірочками). Пароль може і не знадобитися, якщо нещодавно ви вже виконували якусь команду використовуючи sudo²³.



Завдання №2.2.19.

1) Засобами командного рядка спробуйте переглянути вміст файлу /etc/sudoers, не використовуючи sudo. Чи вдається це зробити?

2) Повторіть спробу, додавши команду sudo. Чи вдалося тепер? Чому?²⁴

Рис. 3.5. Методичний посібник «Операційні системи та системне програмування» для бакалаврів інформатики: маркування блоків, які стосуються GUI / CLI

Розмежованість блоків матеріалу, присвячених різним ОС. Коли у межах однієї лабораторної роботи наводяться відомості про роботу з різними ОС, відповідні блоки також мають маркування (рис. 3.6).

Теоретичні відомості та завдання до лабораторної роботи

🔗 Моніторинг пам'яті та часу ЦП у Linux

У деяких варіантах Ubuntu є GUI-засоби для моніторингу пам'яті та процесорного часу (наприклад, у стандартному варіанті Ubuntu з графічним інтерфейсом Unity є програма Системний монітор, який дозволяє як працювати з процесами, так і здійснювати моніторинг використання пам'яті і часу ЦП).

Однак такі GUI-засоби є не в усіх інтерфейсах. Зокрема, у Xubuntu з інтерфейсом Xfce відповідний інструмент за замовчуванням відсутній. Тому у цій роботі розглянемо моніторинг пам'яті та часу ЦП засобами командного рядка.

🖥️ Команда free

Виводить відомості про використання пам'яті.

Вжита без параметрів, команда здійснить виведення, подібне до наступного:

```
student@lab01:~$ free
              total        used        free      shared    buffers     cached
Mem:      250636      239272       11364          0        32280       79332
-/+ buffers/cache:      127660      122976
Swap:      214008       45580      168428
```

Пояснення до стовпчиків

total – загальний обсяг пам'яті (цей обсяг пам'яті буде дещо меншим за той, який ви запалили під час встановлення ОС оскільки частина пам'яті

🖥️ Моніторинг пам'яті та часу ЦП у Windows

Опишемо моніторинг пам'яті та часу ЦП засобами програми Task manager (Диспетчер задач).

Зауваження. Task Manager (Диспетчер задач) у Windows змінювався від версії до версії.

У цій лабораторній роботі за основу взято інтерфейс Диспетчера задач у Windows Server 2012 R2⁴⁸ (набір інструментів **а**) та Windows XP (набір інструментів **б**). Вивчаючи теоретичні відомості, вибирайте опис Диспетчера задач, ближчий до того, з яким ви працюєте, і звертайте увагу на позначки зліва від заголовку завдання:

🖥️ – Windows Server 2012 R2, 🖥️ – Windows XP.

Основне призначення Диспетчера задач – робота з процесами (див. лабораторна робота №6), однак у ньому також є окрема вкладка, присвячена пам'яті – Performance (Продуктивність, Быстродействие).

Рис. 3.6. Методичний посібник «Операційні системи та системне програмування» для бакалаврів інформатики: маркування блоків, які стосуються різних ОС

Методичний посібник містить окремий розділ «Навігатор» у якому пояснюються наведені вище особливості посібника, включно з діаграмою зв'язків між лабораторними та самостійними роботами (рис. 3.7). Якщо для успішного виконання деякої лабораторної роботи треба спершу опрацювати іншу лабораторну роботу (роботи), то це відображається на діаграмі тонкими стрілками (суцільними, пунктирними або штрих-пунктирними). Світлі товсті стрілки показують загальний порядок виконання робіт.

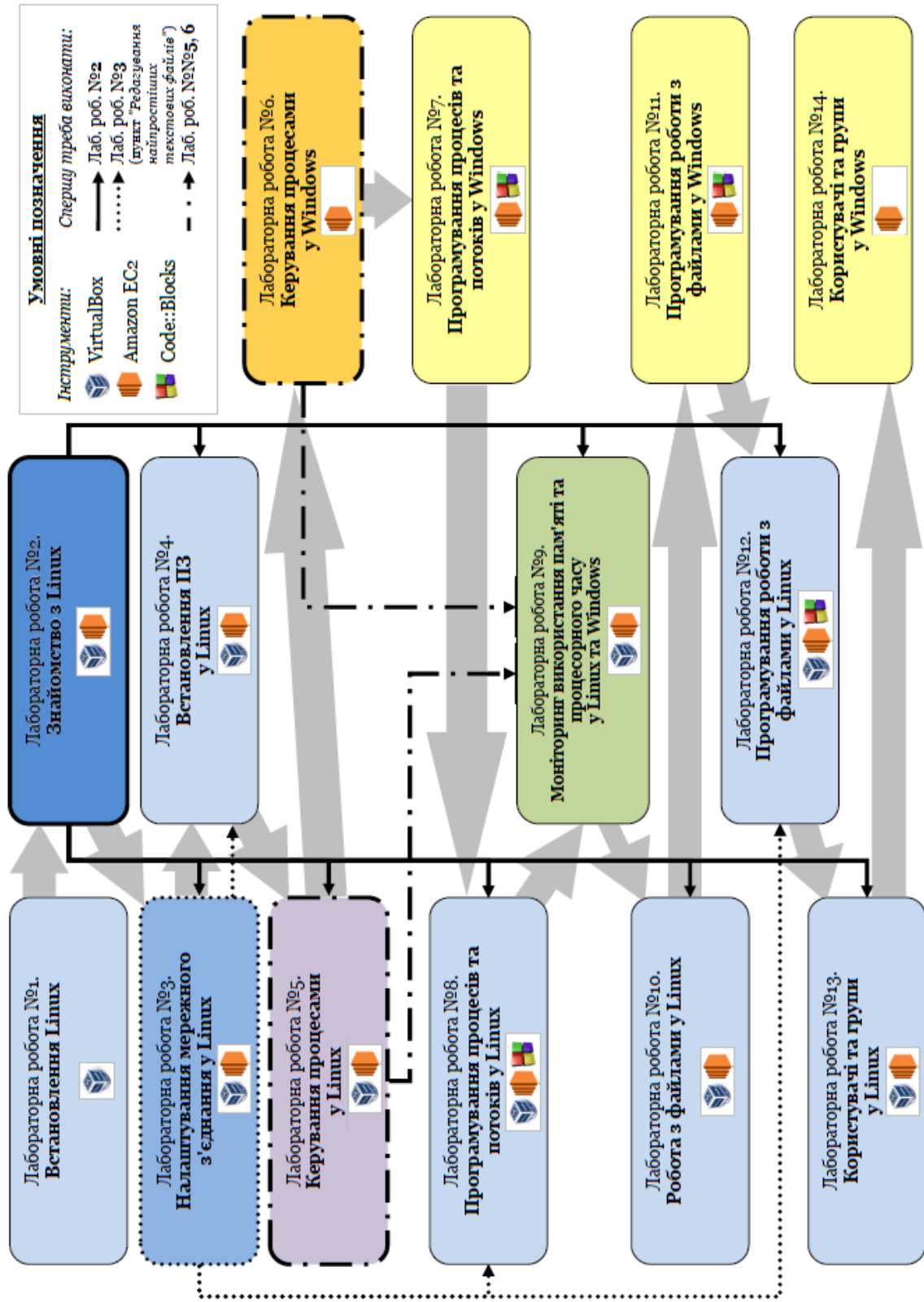


Рис. 3.7. Методичний посібник «Операційні системи та системне програмування» для бакалаврів інформатики: діаграма зв'язків між лабораторними роботами та інструментами

Деякі роботи є ключовими, оскільки на них спирається значна кількість робіт (наприклад, лабораторна робота №2 «Знайомство з Linux» є основою для більшості робіт, присвячених Linux), інші роботи, навпаки, можуть бути вилючені з розгляду без суттєвого впливу на решту робіт (як, наприклад, лабораторна робота №1 «Встановлення Linux» у разі надання студентам готового образу віртуальної машини з Linux, використання віртуальних машин у хмарі на Amazon EC2 тощо). Застосовувані засоби віртуалізації та інші програмні засоби, а також варіанти без віртуалізації показано спеціальними піктограмами біля відповідної лабораторної роботи. На початку кожної роботи, окрім її мети, вказано зв'язки між даною роботою та іншими лабораторними та самостійними роботами (секція «Спершу варто виконати»), засоби віртуалізації та інші засоби, рекомендовані для виконання роботи (секція «Інструменти») – рис. 3.8.

<i>Інструменти:</i>	а) <i>(графічний інтерфейс)</i> VirtualBox + віртуальна машина зі встановленою Xubuntu 14.04 LTS + (рекомендовано) Інтернет-з'єднання; б) <i>(лише командний рядок)</i> VirtualBox + віртуальна машина зі встановленою Ubuntu 14.04 LTS без GUI + (рекомендовано) Інтернет-з'єднання; в) <i>(лише командний рядок)</i> Amazon EC2 + віртуальна машина EC2 t2.micro зі встановленою Ubuntu Server 14.04 LTS + (обов'язково) Інтернет-з'єднання.
<i>Спершу варто виконати:</i>	Самостійна робота №1. Основи роботи з VirtualBox^{а,б}. Самостійна робота №2. Основи роботи з Amazon EC2^в. Лабораторна робота №1. Встановлення Linux. Лабораторна робота №2. Знайомство з Linux.

*Рис. 3.8. Методичний посібник «Операційні системи та системне програмування» для бакалаврів інформатики:
 секції «Інструменти» та «Спершу варто виконати»
 на початку лабораторної роботи*

Перелік рекомендованих засобів згруповано за варіантами – а), б), в) і т.д. Коло засобів, які потенційно можуть бути застосовані під час опрацювання лабораторної роботи, певна річ, є значно ширшим. Запропонований на початку кожної роботи перелік засобів містить саме засоби із описаної раніше

комбінації засобів віртуалізації, а ОС, їх дистрибутиви, версії, випуски, вказані у переліку, було використано під час складання теоретичних відомостей та завдань до лабораторної роботи. Втім, більшість робіт можуть бути повністю чи частково виконані і на інших дистрибутивах, версіях, випусках ОС.

Враховуючи описані вище особливості, викладач може самостійно визначати, чи використовувати засоби віртуалізації в окремій роботі, і якщо так, то які саме. Він також має змогу обирати, які роботи виконувати, а які ні, й послідовність виконання.

Кожна лабораторна робота завершується секціями «Вимоги до захисту лабораторної роботи» (рис. 3.9) та «Контрольні запитання» (рис. 3.10), у яких перераховуються основні результати навчання, що очікуються від студентів після завершення ними виконання даної роботи.

Вимоги до захисту лабораторної роботи

Для успішної здачі лабораторної роботи вам необхідно:

1) вміти:

-  видобувати файли з архіву за допомогою команди `tar` (у межах відомостей, поданих у даній лабораторній роботі);
-  за допомогою команди `dpkg` встановлювати пакунки, деінстановувати пакунки (часткове та повне вилучення), виводити список встановлених пакунків (у тому числі, відфільтрований за ім'ям пакунку чи фрагментом імені пакунку);
-  за допомогою команди `uname` виводити відомості про ядро ОС та аналізувати вивід цієї команди з метою визначення розрядності ОС;
-  за допомогою команди `wget` завантажувати файл, знаючи посилання на нього (у межах відомостей, поданих у даній лабораторній роботі);
- аналізувати вміст файлу `/etc/apt/sources.list` (чи аналогічних файлів);
-  шляхом редагування файлу `/etc/apt/sources.list` включати та відключати описані там репозиторії;
-  за допомогою команди `apt-cache` здійснювати пошук пакунків у підключених репозиторіях, виводити відомості про пакунок за його ім'ям;
-  за допомогою команди `apt-get` встановлювати пакунки, деінстановувати пакунок (частково та повністю), здійснювати автоматичне вилучення пакунків, встановлених разом з іншими пакунками і наразі більше не потрібних, а також автоматичне очищення системи від `deb`-файлів уже деінстановованих пакунків;
-  запускати менеджер пакунків Synaptic з правами адміністратора;
-  у менеджері пакунків Synaptic розрізняти встановлені пакунки, пакунки, позначені для встановлення, для часткового вилучення, для повного вилучення;

*Рис. 3.9. Методичний посібник «Операційні системи та системне програмування» для бакалаврів інформатики:
вимоги до захисту лабораторної роботи*

Контрольні запитання

- 1) Поясніть суть поняття *пакунок*. Що входить до складу пакунку?
- 2) Якими способами можуть поширюватися пакунки?
- 3) Що таке репозиторій?
- 4) Що у випадку пакунків означають позначки *binaries* та *sources*?
- 5) Назвіть відомі вам формати пакунків. Який із цих форматів використовується в Ubuntu?
- 6) Поясніть поняття *залежностей* між паунками. Що означає *автоматичне врахування залежностей*?
- 7) Перерахуйте відомі вам інструменти для роботи з deb-паунками. Чим вони відрізняються між собою? У яких випадках рекомендується використання кожного з цих інструментів?
- 8) Чи може програмне забезпечення для Linux поширюватися без використання пакунків? Як у такому разі відбувається його встановлення?
- 9) Назвіть чотири компоненти (області), за якими організовані репозиторії Ubuntu. Поясніть, чим вони відрізняються один від одного.
- 10) Перерахуйте чотири категорії, на які поділяються оновлення для Ubuntu, та коротко охарактеризуйте кожну з цих категорій.
- 11) У якому файлі найчастіше зберігається перелік репозиторіїв, з якими працює APT? Де іще може зберігатися цей перелік?

*Рис. 3.10. Методичний посібник «Операційні системи та системне програмування» для бакалаврів інформатики:
контрольні запитання для захисту лабораторної роботи*

Робота завершується секцією «Джерела та посилання», котра містить матеріали, які було використано під час створення роботи, та які рекомендуються як основа для наступного поглиблення знань з теми (рис. 3.11).

Джерела та посилання

1. Пакетная система: Инструменты и основные принципы. – Настольная книга администратора Debian. – Глава 5. – Режим доступа: <https://debian-handbook.info/browse/ru-RU/stable/sect.manipulating-packages-with-dpkg.html> (рус. + eng.)
2. Ubuntu Manuals. – Access mode: <http://manpages.ubuntu.com/manpages/trusty/en/man5/interfaces.5.html> (eng.)
3. Repositories/Ubuntu. – Ubuntu Community Help Wiki. – Access mode: <https://help.ubuntu.com/community/Repositories/Ubuntu> (eng.)
4. Repositories/CommandLine. – Ubuntu Community Help Wiki. – Access mode: <https://help.ubuntu.com/community/Repositories/CommandLine> (eng.)
5. Package Management / Ubuntu 14.04. Настанова користувача Ubuntu Сервер. – Ubuntu Documentation: Official Documentation. – Access mode: <https://help.ubuntu.com/lts/serverguide/package-management.html> (eng.)
6. AptGet/Howto. – Ubuntu Community Help Wiki. – Access mode: <https://help.ubuntu.com/community/AptGet/Howto?action=show&redirect=AptGetHowto> (eng.)

*Рис. 3.11. Методичний посібник «Операційні системи та системне програмування» для бакалаврів інформатики:
перелік джерел та посилань у кінці лабораторної роботи*

Зазначені у секції «Вимоги до захисту лабораторної роботи» уміння, як і блоки основної частини лабораторної роботи, мають спеціальне маркування залежно від того, який інтерфейс застосовується для вироблення та демонстрації цього вміння – графічний чи інтерфейс командного рядка. Якщо у деякій лабораторній роботі йдеться про кілька ОС, то відповідні уміння також марковано. Контрольні запитання спрямовані на з'ясування, наскільки усвідомленим є використання студентом команд, операцій тощо, на виявлення його розуміння переваг та обмежень описаних у роботі засобів та технологій, орієнтування у доступних альтернативах. Контрольні запитання також можуть мати маркування, аналогічне до використовуваного у секції «Вимоги до захисту лабораторної роботи».

Разом секції «Вимоги до захисту лабораторної роботи» та «Контрольні запитання» відповідають сформульованим у п. 3.2 результатам навчання. Зауважимо однак, що очікувані результати навчання не зводяться лише до тих, які мають бути одержані у процесі виконання лабораторних робіт, оскільки значна частина цих результатів (передусім, але не виключно – підпорядкованих завданню (1) (навчання основ теорії ОС) досягається під час проходження теоретичної частини курсу (лекції, контрольні тестові опитування, екзамен).

Посилання для секції «Джерела і посилання» дібрано, зважаючи передусім на авторитетність відповідних джерел. Секція містить відомі підручники та керівництва з ОС, офіційні довідкові ресурси.

Виконання студентами практичних завдань може здійснюватися як у комп'ютерних лабораторіях ЗВО, так і поза університетом, причому в обох випадках допускається робота зі щонайменше двох класів пристроїв: з настільних ПК та з ноутбуків. У разі залучення хмарних сервісів також можливе застосування мобільних пристроїв (планшети, смартфони тощо) однак наразі це питання нами не досліджувалося і потребує окремого вивчення.

Різноманітність потенційно доступних апаратних конфігурацій, а також притаманні їм переваги та обмеження впливають на вибір засобів віртуалізації. Так, під час впровадження пропонованої методики у Житомирському

державному університеті імені Івана Франка настільні ПК у комп'ютерних лабораторіях поступалися ноутбукам студентів за швидкістю роботи. Хмарні сервіси не мають таких обмежень, однак університетська мережа у моменти пікових навантажень не завжди забезпечувала пропускну здатність, достатню для ефективної роботи з віртуальними машинами у хмарі. Не всі студенти мали змогу приходити на заняття з ноутбуками, не у всіх були сприятливі умови для роботи з віртуалізованою ОС поза університетом (недостатньо потужне апаратне забезпечення, відсутність стабільного доступу до Інтернету тощо). Однак у межах варіативного підходу до застосування засобів віртуалізації кожний студент одержував доступ до віртуалізованої ОС за допомогою одного чи кількох засобів віртуалізації або працював з невіртуалізованою ОС, якщо висловлював таке бажання (Linux як друга ОС на студентському ноутбуці). Крім того, абстрагування від конкретних засобів віртуалізації й конкретного інтерфейсу надає студентам простір для вироблення власних оригінальних рішень, початково відсутніх серед запропонованих (наприклад, основна ОС Mac OS X та гіпервізор Parallels Desktop).

Завдання 1 пропонованого варіативного підходу полягає в ознайомленні викладачів та студентів з різноманітністю засобів віртуалізації. Для усвідомленої реалізації пропонованої методики застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики на практиці важливе розуміння викладачем суті технологій віртуалізації, різноманітності засобів, доступних для використання у курсі з ОС, їх відмінностей, можливостей та обмежень. Це дасть змогу, по-перше, не лише застосовувати засоби віртуалізації із описаної вище комбінації засобів віртуалізації, а й самостійно добирати засоби віртуалізації для курсу з ОС відповідно до індивідуальних потреб, й по-друге, сформувати відповідні знання, уміння, навички, ставлення, досвід щодо технологій віртуалізації у студентів. У даному дослідженні ми розглядаємо технології віртуалізації передусім як засіб навчання, а тому нижче йтиметься саме про ознайомлення викладачів з технологіями віртуалізації.

Результатом реалізації *завдання 1* описуваного варіативного підходу стали методичні рекомендації «Технології віртуалізації у навчанні операційних систем бакалаврів інформатики» [50].

З практичної точки зору, методичні рекомендації «Технології віртуалізації у навчанні операційних систем бакалаврів інформатики» спираються на наступні твердження:

1) вибір засобів віртуалізації для курсу з ОС має бути індивідуальним та усвідомленим;

2) існують підстави та можливості для поєднання кількох засобів віртуалізації у межах одного курсу з ОС.

Методичні рекомендації «Технології віртуалізації у навчанні операційних систем бакалаврів інформатики» містять відомості про:

- різноманітність тлумачень терміну «віртуалізація», зокрема в інформатиці;
- зв'язок віртуалізації та освіти, аспекти цього зв'язку та його інтерпретацію з точки зору теорії множин;
- коротку історію розвитку технологій віртуалізації;
- систематизацію технологій віртуалізації;
- хмарні технології та їх зв'язок з технологіями віртуалізації;
- основні ідеї та положення варіативного підходу до застосування віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики;
- приклад поєднання кількох засобів віртуалізації в одному курсі з ОС;
- засоби віртуалізації, які можуть бути використані у курсі з ОС і які належать до різних категорій таких засобів із зазначенням основних переваг та недоліків;
- рекомендації щодо добору засобів віртуалізації для їх наступного використання у курсі з ОС для бакалаврів інформатики;
- деякі публікації, рекомендовані для самостійного опрацювання (перелік структурований і супроводжується короткими коментарями).

Огляд засобів віртуалізації містить опис прикладів засобів віртуалізації, які належать до різних напрямів віртуалізації, спираються на різні методи віртуалізації чи їх поєднання. Розглянуто як засоби віртуалізації із комбінації засобів віртуалізації (Oracle VirtualBox, Amazon EC2), так і деякі інші засоби, відсутні в основній комбінації, але можуть бути використані у межах варіативного підходу до застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики (Proxmox VE, Canonical LXD, IDE Cloud9). Подано базові технічні відомості, правову інформацію про описувані засоби, виокремлено головні переваги та недоліки кожного з них. У методичних рекомендаціях наголошено на тому, що включені до видання засоби віртуалізації наведені для прикладу, і їх може бути замінено іншими засобами з подібним функціоналом.

Важливою частиною методичних рекомендацій «Технології віртуалізації у навчанні операційних систем бакалаврів інформатики» є поради щодо добору засобів віртуалізації. Пропоновані категорії засобів віртуалізації для Linux із відповідними варіантами застосування технологій віртуалізації для навчання Windows, подано у таблиці 3.3.

Таблиця 3.3

Засоби віртуалізації відповідно до варіативного підходу до застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики

<i>Варіанти з Windows</i>	<i>Варіанти з Linux</i>
1) Умовно безкоштовна віртуальна машина з Windows	1) Гіпервізор II типу (локально) (Oracle VirtualBox, VMware Player, KVM та ін.)
2) Віртуальні машини з Windows у хмарі (Amazon EC2, Google Cloud Platform, Microsoft Azure)	2) Готові віртуальні машини (віртуальні контейнери) у хмарі (Amazon EC2, Google Cloud Platform, Cloud9 та ін.)
3) Windows без віртуалізації	3) Віртуальні контейнери (локально) (OpenVZ, FreeBSD jail, Linux-Vserver, LXC, Canonical LXD та ін.)
	4) Сервер з віртуальними машинами/контейнерами (Proxmox VE; Oracle VM Server for x86, XenServer; Docker, OpenVZ, FreeBSD jail, Linux-Vserver, LXC, Canonical LXD; Apache CloudStack, Eucalyptus, OpenStack та ін.)

Також у методичних рекомендаціях запропоновано процедуру добору засобів віртуалізації для навчання ОС бакалаврів інформатики (рис. 3.12).

Процедура має рекомендаційний характер і дає змогу зорієнтуватися на початку пошуку, виключити з подальшого розгляду засоби, які в окремо взятому випадку підходять менше за інші.

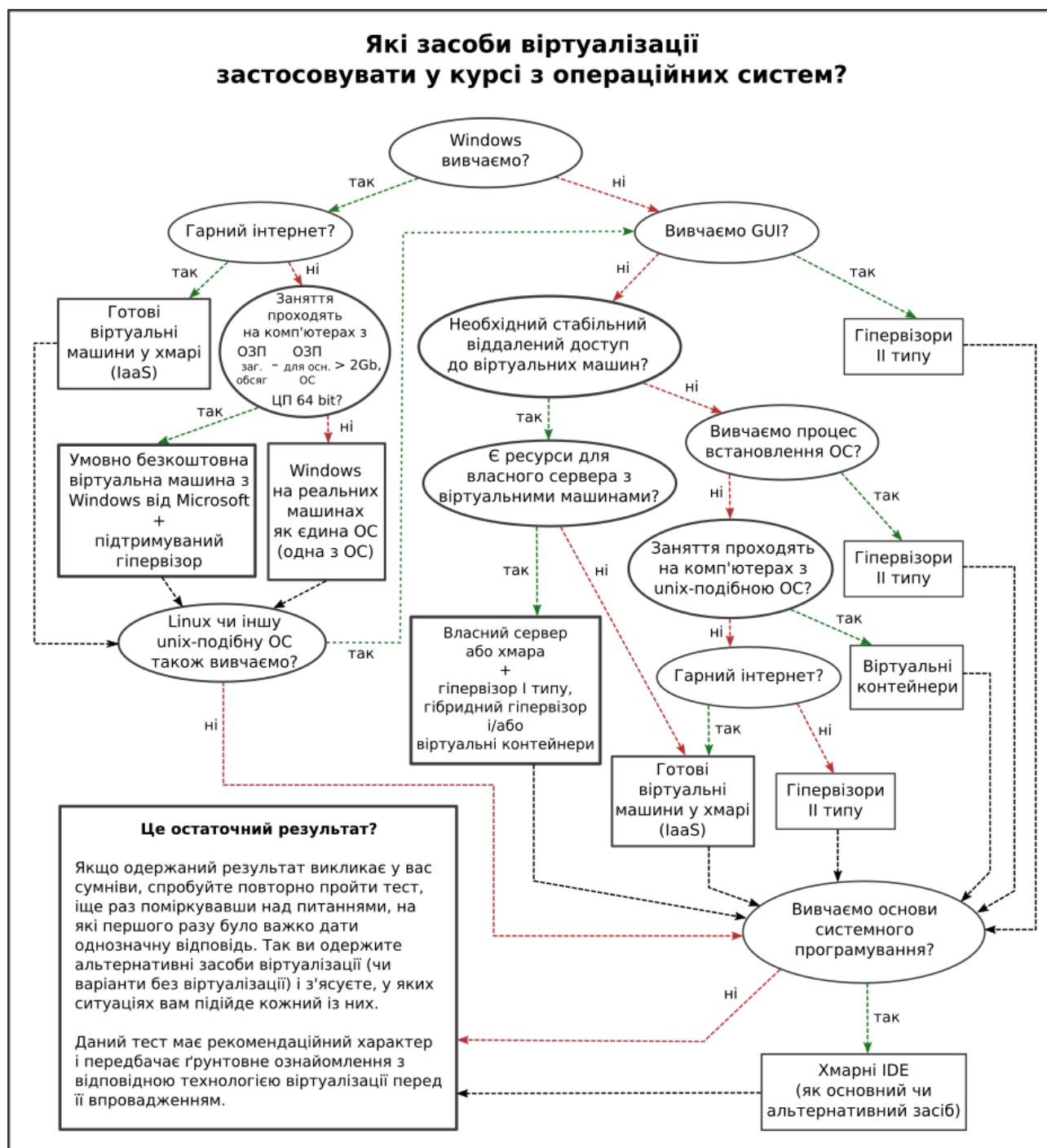


Рис. 3.12. Методичні рекомендації «Технології віртуалізації у навчанні операційних систем бакалаврів інформатики»: процедура добору засобів віртуалізації для навчання ОС бакалаврів інформатики

3.4. Зміст, форми, методи і засоби навчання під час застосування технологій віртуалізації Unix-подібних операційних систем у межах змістового модуля «Файлові системи»

Детальніше опишемо застосування технологій віртуалізації Unix-подібних ОС у межах окремого змістового модуля.

Майже всі змістові модулі курсу з ОС допускають активне застосування засобів віртуалізації, коли модулем передбачено виконання лабораторних робіт. Наявність лабораторних робіт у змістовому модулі та їх зміст може відрізнятися для окремо взятого курсу. Відомості про змістові модулі, які містять лабораторні роботи згідно з пропонованою методикою застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики, наведено у таблиці 3.4, й у такому разі йдеться про п'ять змістових модулів з восьми.

Таблиця 3.4

Перелік змістових модулів дисципліни «Операційні системи та системне програмування» та відповідних лабораторних робіт згідно з пропонованою методикою

Назва модуля	Лабораторні роботи
Змістовий модуль 1. Початкові відомості про ОС	Лабораторна робота №1. Встановлення Linux Лабораторна робота №2. Знайомство з Linux Лабораторна робота №3. Налаштування мережного з'єднання у Linux Лабораторна робота №4. Встановлення ПЗ у Linux
Змістовий модуль 2. Процеси та потоки	Лабораторна робота №5. Керування процесами у Linux Лабораторна робота №6. Керування процесами у Windows Лабораторна робота №7. Програмування процесів та потоків у Windows Лабораторна робота №8. Програмування процесів та потоків у Linux
Змістовий модуль 3. Пам'ять	Лабораторна робота №9. Моніторинг використання пам'яті та процесорного часу у Linux та Windows
Змістовий модуль 4. Файлові системи	Лабораторна робота №10. Робота з файлами у Linux Лабораторна робота №11. Програмування роботи з файлами у Windows Лабораторна робота №12. Програмування роботи з файлами у Linux

<i>Назва модуля</i>	<i>Лабораторні роботи</i>
<i>Змістовий модуль 5. Безпека</i>	Лабораторна робота №13. Користувачі та групи у Linux Лабораторна робота №14. Користувачі та групи у Windows
<i>Змістовий модуль 6. Планування</i>	Лабораторних робіт не передбачено
<i>Змістовий модуль 7. Міжпроцесова та міжпоточкова взаємодія</i>	Лабораторних робіт не передбачено
<i>Змістовий модуль 8. Віртуалізація</i>	Лабораторних робіт не передбачено

З метою вибору змістового модуля для детального розгляду на його прикладі запропонованої методики застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики, та, зокрема, варіативного підходу до використання таких технологій, проведено експертне опитування (додаток М).

Опитування було організоване у формі електронного анкетування. У ролі експертів залучено представників закладів вищої освіти України (разом 15 респондентів). Більшість учасників опитування (14 осіб, або 93,3%) мали досвід викладання курсу з ОС студентам педагогічних спеціальностей, причому двоє учасників (13,3%) також відзначили свій досвід викладання відповідної дисципліни студентам інженерних спеціальностей, один учасник (6,7%) – досвід викладання студентам правничих спеціальностей й один учасник (6,7%) – досвід написання підручників для закладів вищої освіти з інфокомунікацій, ОС та ІКТ.

Експертам пропонувалося висловити свою думку щодо пріоритетності змістових модулів курсу з ОС. Запитання мало наступне формулювання: «Оцініть важливість запропонованих тем курсу з операційних систем для бакалаврів інформатики (майбутні вчителі інформатики)». Для оцінювання було запропоновано наступний перелік змістових модулів.

- 1) Початкові відомості про операційні системи.
- 2) Процеси та потоки.
- 3) Пам'ять.
- 4) Файлові системи.
- 5) Безпека.

- 6) Планування.
- 7) Міжпроцесова та міжпотоківна взаємодія.
- 8) Віртуалізація.
- 9) Керування пристроями.

Оцінювання здійснювалося за трибальною шкалою від 1 до 3 (1 – можна розглядати, 2 – варто розглядати, 3 – обов'язково розглядати). Одержані результати наведено у табл. 3.5.

Таблиця 3.5

**Результати оцінювання експертами
запропонованого переліку змістових модулів курсу з ОС**

Модуль	Оцінки															Сума оцінок
	Експерт 1	Експерт 2	Експерт 3	Експерт 4	Експерт 5	Експерт 6	Експерт 7	Експерт 8	Експерт 9	Експерт 10	Експерт 11	Експерт 12	Експерт 13	Експерт 14	Експерт 15	
Початкові відомості про ОС	3	3	3	3	3	3	1	3	3	2	3	3	3	3	2	41
Процеси та потоки	2	3	2	2	3	2	2	2	2	2	3	2	3	2	3	35
Пам'ять	2	3	2	2	3	2	2	3	3	3	3	3	3	3	3	40
Файлові системи	3	3	3	3	3	2	2	3	3	3	3	3	3	3	3	43
Безпека	3	3	3	3	3	3	3	3	3	1	3	3	3	3	3	43
Планування	2	2	3	3	2	2	1	2	2	1	3	2	3	2	2	32
Міжпроцесова та міжпотоківна взаємодія	1	1	2	2	2	1	2	2	3	2	3	2	3	1	1	28
Віртуалізація	2	1	3	3	2	1	3	2	2	2	3	2	3	1	2	32
Керування пристроями	3	3	3	3	2	2	2	3	3	3	2	2	2	2	2	37

У процесі попереднього аналізу даних табл. 3.5 виникає припущення про досить невисоку однотайність експертів: кожний із запропонованих змістових модулів має оцінки щонайменше за двома різними рівнями, у тому числі за рівнями, які в обраній шкалі є мінімумом і максимумом (1 і 3).

Для визначення ступеню узгодженості оцінок експертів щодо пріоритетності змістових модулів курсу з ОС для бакалаврів інформатики було обчислено коефіцієнт узгодженості Кендалла.

Коефіцієнт узгодженості Кендалла (Kendall's coefficient of concordance, W) є мірою узгодженості між m змінними, значення яких є оцінками кожного з

n об'єктів [243]. В аналізованому випадку у ролі змінних виступають опитані експерти ($m = 15$), а у ролі об'єктів – оцінювані змістові модулі курсу з ОС ($n = 9$). Значення коефіцієнта узгодженості Кендалла змінюється у межах від 0 до 1, причому значення 0 відповідає відсутності узгодженості між експертами, а значення 1 – повній однаковості експертів.

Одержаним даним притаманні так звані зв'язані ранги: серед оцінок, які відповідають одному об'єкту, повторюються однакові значення. Наприклад, для першого об'єкта (змістового модуля), маємо повторювані значення 3 і 2: {3; 3; 3; 3; 3; 3; 1; 3; 3; 2; 3; 3; 3; 3; 2}. Отже, має застосовуватися формула для обчислення коефіцієнта узгодженості Кендалла з коригуванням зв'язаних рангів [243]:

$$W = \frac{12S}{m^2(n^3 - n) - mT}, \quad (3.1)$$

де

$$S = \sum_{i=1}^n (R_i - \bar{R})^2, \quad (3.2)$$

$$T = \sum_{k=1}^g (t_k^3 - t_k), \quad (3.3)$$

де t_k – кількість зв'язаних рангів у k -й групі зв'язаних рангів, а g – кількість таких груп.

Засобами мови статистичного виведення R (функція *KendallW* з бібліотеки *DescTools*) обчислено значення коефіцієнта узгодженості Кендалла $W = 0,33$, що свідчить про невисоку однаковість експертів. Перевірка значущості за критерієм хі-квадрат за ступенями вільності кількістю $n - 1 = 8$ показала статистичну значущість одержаного результату на рівні значущості 0,01 (одержане p -значення $p\text{-value} = 0,0000036 < 0,01$), що також підтверджується перевіркою значущості за перестановочним тестом (permutation test) на основі функції *kendall.w* з бібліотеки *synchrony* з кількістю перестановок 100000 ($p\text{-value} = 0,00002 < 0,01$). Програмні коди мовою R та відповідні результати наведено у додатку Н.

Невисоку однаковість експертів могли спричинити, зокрема такі фактори, як:

- відсутність на момент проведення опитування нового затвердженого галузевого стандарту вищої освіти України для спеціальностей, які передбачають підготовку вчителів інформатики;

- недостатня розробленість переліку компетентностей, пов'язаних з ОС, враховуючи інформатичні компетентності бакалаврів інформатики щодо ОС;

- можливість віднесення низки питань до різних змістових модулів;

- вплив міжпредметних зв'язків курсу з ОС (деякі змістові модулі можуть розглядатися у межах інших дисциплін).

Втім, навіть враховуючи розходження поглядів експертів загалом, щодо окремих змістових модулів оцінки експертів виявилися суттєво вищими (табл. 3.6).

Таблиця 3.6

**Виокремлення найвищих суми оцінок, даних експертами
змістовим модулям курсу з ОС із запропонованого переліку**

Модуль	Оцінки															Сума оцінок
	Експерт 1	Експерт 2	Експерт 3	Експерт 4	Експерт 5	Експерт 6	Експерт 7	Експерт 8	Експерт 9	Експерт 10	Експерт 11	Експерт 12	Експерт 13	Експерт 14	Експерт 15	
Початкові відомості про ОС	3	3	3	3	3	3	1	3	3	2	3	3	3	3	2	41
Процеси та потоки	2	3	2	2	3	2	2	2	2	2	3	2	3	2	3	33
Пам'ять	2	3	2	2	3	2	2	3	3	3	3	3	3	3	3	40
Файлові системи	3	3	3	3	3	2	2	3	3	3	3	3	3	3	3	43
Безпека	3	3	3	3	3	3	3	3	3	1	3	3	3	3	3	43
Планування	2	2	3	3	2	2	1	2	2	1	3	2	3	2	2	32
Міжпроцесова та міжпотоків взаємодія	1	1	2	2	2	1	2	2	3	2	3	2	3	1	1	28
Віртуалізація	2	1	3	3	2	1	3	2	2	2	3	2	3	1	2	32
Керування пристроями	3	3	3	3	2	2	2	3	3	3	2	2	2	2	1	37

Серед чотирьох модулів, сума оцінок яких найбільша: «Файлові системи» (сума 43, 13 оцінок «3», 2 оцінки «2»), «Безпека» (сума 43, 14 оцінок «3», 1 оцінка «1»), «Початкові відомості про операційні системи» (сума 41, 13 оцінок «3», 2 оцінки «2», 1 оцінка «1»), «Пам'ять» (сума 40, 10 оцінок «3»,

5 оцінок «2»). Водночас варто відзначити, що сума оцінок решти модулів не є суттєво нижчою за 40 й варіює у межах від 28 до 37, а модуль, який би набрав переважно оцінки «1» («можна вивчати»), за результатами опитування, відсутній.

Розглядатимемо зміст, форми, методи і засоби навчання під час застосування технологій віртуалізації Unix-подібних ОС на прикладі змістового модуля «Файлові системи» (найвища загальна сума балів, жодної оцінки «1», а також передбаченість модулем лабораторних робіт згідно з пропонованою методикою). Водночас, не заперечуємо важливості інших змістових модулів, як і можливості під час їх вивчення за потреби застосовувати технології віртуалізації.

Зміст, форми, методи і засоби навчання під час застосування технологій віртуалізації Unix-подібних ОС у межах змістового модуля «Файлові системи» відповідають змістовому (зміст) та технологічному (форми, методи і засоби) компонентам моделі.

Зміст навчання. Відповідно до пропонованої методики, змістовим модулем «Файлові системи» передбачено лекції та лабораторні роботи. Питання за матеріалом даного змістового модуля також входять до другого з двох передбачених курсом контрольних тестових опитувань з теорії ОС, яке проводиться згодом.

Опис лекційної частини змістового модуля наведено у таблиці 3.7.

Таблиця 3.7

**Опис лекційної частини змістового модуля «Файлові системи»
згідно з пропонованою методикою**

<i>Характеристика</i>	<i>Опис характеристики</i>
Кількість аудиторних годин (орієнтовно)	2
Питання, що розглядаються	Файл. Особливості іменування файлів. Файлові атрибути. Каталог як група файлів. Каталог як спеціальний файл. Абсолютне та відносне ім'я файлу. Дискові розділи. Огляд конкретних файлових систем (FAT, NTFS, UFS, ext). Сумісність файлових систем

<i>Характеристика</i>	<i>Опис характеристики</i>
Результати навчання	Знати типові файлові атрибути. Уміти порівнювати правила іменування файлів у файлових системах для Unix-подібних ОС та файлових системах для Windows. Уміти описувати призначення дискових розділів та їх роль у завантаженні ОС. Уміти характеризувати основи будови та функціонування конкретних файлових систем (FAT, NTFS, UFS, ext). Уміти обґрунтовувати потребу та пояснювати суть технології журналювання. Уміти пояснювати роль віртуальних файлових систем у реалізації сумісності файлових систем.
Ключові поняття	Файл; атрибут; каталог; абсолютне та відносне ім'я; файлова система; журналювання; віртуальна файлова система.
Зв'язок зі змістовими модулями CS2013	ОС/Файлові системи, ОС/Безпека і захист, ОС/Віртуальні машини

Змістовий модуль містить три лабораторні роботи: «Робота з файлами у Linux», «Програмування роботи з файлами у Windows», «Програмування роботи з файлами у Linux». Кількість робіт та склад робіт можуть коригуватися у разі зміни загальної кількості годин, відведеної на вивчення дисципліни. У зв'язку з цим наведені далі обсяги аудиторних годин для виконання кожної лабораторної роботи є орієнтовними.

Лабораторна робота №10. Робота з файлами у Linux (4 години)

Потребує теоретичних відомостей зі змістових модулів: «Початкові відомості про операційні системи», «Процеси та потоки», «Пам'ять», «Файлові системи».

Результати навчання:

- уміти здійснювати навігацію каталогами та найпростіші операції з файлами у графічному файловому менеджері (лише якщо вивчається GUI);
- уміти здійснювати навігацію каталогами за допомогою команди cd;
- уміти виводити вміст каталогу, зокрема відформатований та відфільтрований, за допомогою команди ls;

- уміти створювати порожній файл засобами команди touch;
- уміти створювати нові каталоги за допомогою команди mkdir;
- уміти здійснювати копіювання, переміщення, перейменування, видалення файлів та каталогів за допомогою команд cp, mv, rm;
- уміти пояснювати суть механізму монтування;
- уміти здійснювати автоматичне/ручне монтування та розмонтування (на прикладі флеш-носія, спільного каталогу віртуальної машини тощо);
- уміти інтерпретувати вміст каталогу /dev;
- уміти одержувати та аналізувати відомості про використання дискового простору файловою системою (команда df);
- уміти виводити та аналізувати таблицю розділів за допомогою команди fdisk;
- уміти створювати файлові архіви у форматах gzip та bzip2, переглядати вміст цих архівів та здійснювати їх розархівування;
- уміти пояснювати відмінність між блоковими та символьними пристроями;
- уміти описувати етапи, з яких складається процес архівування у Linux;
- знати найвідоміші формати файлових архівів у Linux та окреслювати коло ситуацій, коли використання кожного із них доцільне.

Засоби віртуалізації та інші ПЗ: Oracle VirtualBox (основний); Amazon EC2; без віртуалізації.

Лабораторна робота №11. Програмування роботи з файлами у Windows (4 години)

Потребує теоретичних відомостей зі змістових модулів: «Початкові відомості про операційні системи», «Процеси та потоки», «Пам'ять», «Файлові системи».

Результати навчання:

- уміти здійснювати пошук довідкової інформації щодо використання Win32 API в MSDN Library;

- уміти використовувати API-функції CreateFile(), CloseHandle(), SetFilePointer(), ReadFile(), WriteFile(), GetFileInformationByHandle(), DeleteFile(), CopyFile(), MoveFile(), CreateDirectory(), RemoveDirectory(), GetCurrentDirectory().

Засоби віртуалізації та інше ПЗ: без віртуалізації, Code::Blocks (основні); Amazon EC2, Code::Blocks.

Лабораторна робота №12. Програмування роботи з файлами у Linux (4 години)

Потребує теоретичних відомостей зі змістових модулів: «Початкові відомості про операційні системи», «Процеси та потоки», «Пам'ять», «Файлові системи».

Результати навчання:

- уміти здійснювати пошук довідкової інформації щодо використання бібліотечних функцій стандарту POSIX у документації MAN;
- уміти використовувати функції open(), close(), lseek(), read(), write(), fstat(), rename(), mkdir(), unlink(), rmdir().

Засоби віртуалізації та інше ПЗ: Oracle VirtualBox, g++ (основні); Oracle VirtualBox, g++, Code::Blocks; Amazon EC2, g++.

Форми та методи навчання використовуються традиційні, скомбіновані з урахуванням доцільності їх залучення під час навчання окремих змістових модулів чи окремих видів роботи.

Описуваною методикою передбачено застосування таких **форм організації навчання**, як лекція-бесіда, лекція-презентація, лабораторна робота, аудиторна контрольна робота, екзамен.

Для подання загальних відомостей з теорії ОС використовуються лекція-презентація та лекція-бесіда. Теоретичні відомості, пов'язані з питаннями адміністрування ОС та системного програмування у них, містяться в інструктивно-методичних матеріалах до лабораторних робіт і зазвичай перемежуються практичними завданнями. Діагностування сформованості компетентностей здійснюється у формі захисту студентами лабораторних робіт

(демонстрація практичних умінь, знання відповідей на контрольні запитання), контрольних тестових опитувань з теорії ОС та підсумкового усного екзамену.

Такі **методи навчання**, як розповідь, пояснення, бесіда (словесні) застосовуються передусім під час лекцій. Під час лабораторних занять зі словесних методів використовується інструктаж, пояснення, але переважають наочні (демонстрація, спостереження) та практичні методи (лабораторна робота). Додатково передбачено залучення інтерактивних методів, зокрема «мозкового штурму» та ситуаційного аналізу.

Наприклад, означення файлу формулюємо, попередньо запропонувавши студентам згадати відповідні означення самим, оскільки поняття файлу вводиться у низці інших дисциплін. У відповідях студентів виокремлюємо найсуттєвіші риси. Приміром, так: файл є набором даних; цей набір даних впорядкований; файл обов'язково має ім'я; розглядаємо файли саме у комп'ютерних системах (бо термін «файл» багатозначний). Тоді вводимо означення файлу, наприклад, у наступному формулюванні: «Файлом називається впорядкований набір даних, що зберігається у комп'ютерній системі під спільним ім'ям».

Розгляд типових файлових атрибутів можна розпочати з «мозкового штурму», попередньо уточнивши, що атрибутами називають суттєві властивості, і запропонувавши студентам назвати, які атрибути притаманні файлам. Всі озвучені студентами варіанти записуємо на дошці, після чого обговорюємо одержаний перелік: яких пунктів не вистачає, які – зайві чи дублюють те саме поняття. Скоригований перелік атрибутів розглядаємо детальніше.

Можна поєднати подібні запитання до студентів з використанням засобів віртуалізації. Зокрема, більшість атрибутів відображаються у властивостях файлів, виведених через графічний інтерфейс чи інтерфейс командного рядка. Можна додати у лекцію знімок екрану з кількома такими прикладами й запропонувати їх проаналізувати. Однак якщо виведення файлових атрибутів здійснюватиметься наживо, з запущеної операційної системи, то файл для аналізу зможуть вибрати самі студенти. Проводимо цей експеримент у Linux

(наприклад, у файловій системі Ext4) та у Windows (у файловій системі NTFS). Якщо йдеться, скажімо, про трансляцію експерименту по мережі чи використання проектора, всі маніпуляції з файлами відбуватимуться на одному комп'ютері. Тому незалежно від того, скільки ОС встановлено на комп'ютері, з якого ведеться показ, заради економії часу доцільно вдатися до віртуалізації.

В аналогічний спосіб можна застосовувати засоби віртуалізації під час обговорення відмінностей вимог до імен файлів у різних системах. Спершу поставимо студентам запитання: «Чи розрізняється регістр символів (велика і мала літера) в іменах файлів у Windows? А у Linux?» Почувши припущення студентів, запропонуємо перевірити ці припущення експериментально, спробувавши створити в одному каталозі два файли з іменами, відмінними лише регістром окремих символів.

Інструктивно-методичні матеріали до лабораторних робіт також містять ряд потенційних проблемних ситуацій, які можуть виникати у студента під час виконання практичних завдань, супроводжувані підказками для самостійного пошуку відповідей.

Так, у самостійній роботі №1 «Основи роботи з VirtualBox» у блоці «Взяти образ віртуальної машини» подано відомості про деякі ресурси, що можуть стати додатковими джерелами образів готових віртуальних машин для VirtualBox, поряд із образами, наданими викладачем (VirtualBoxes [286], Turnkey Linux [283], хмарні платформи моделі IaaS на зразок Amazon EC2). Інший блок цієї самостійної роботи «32 біти чи 64 біти» пояснює, як з'ясувати, гостьові ОС якої розрядності може використовувати студент із наявним у нього апаратним забезпеченням та встановленою на ньому основною ОС.

Засоби навчання, використовувані у межах змістового модуля «Файлові системи», співпадають із засобами навчання, застосування яких передбачене описуваною методикою загалом: апаратне і програмне забезпечення (зокрема засоби віртуалізації), підручники і посібники, освітні і професійні веб-ресурси та інші засоби (звичайні та маркерні дошки, роздаткові матеріали тощо).

Детальніше розглянемо застосування засобів віртуалізації відповідно до

пропонованої методики на прикладі змістового модуля «Файлові системи».

Засоби віртуалізації, як і решту засобів навчання, використано у поєднанні з формами організації навчання та методами навчання, спираючись на них і водночас доповнюючи їх і сприяючи їх реалізації (табл. 3.8).

Таблиця 3.8

Використання засобів віртуалізації у поєднанні з формами та методами навчання згідно з пропонованою методикою

Форми	Методи	Суть застосування засобів віртуалізації
Лекція-бесіда, лекція-презентація	Розповідь, пояснення, бесіда	<u>Використовуються епізодично:</u> для показу прикладів обговорюваних ОС у процесі реальної роботи.
Лабораторна робота	Інструктаж, пояснення; демонстрація, спостереження; лабораторна робота	<u>Використовуються активно:</u> для виконання студентами завдань лабораторної роботи та захисту лабораторних робіт. <u>Використовують епізодично:</u> для демонстрації і пояснення початків роботи з засобами віртуалізації (за потреби); для демонстрації і пояснення виконання окремих фрагментів лабораторних робіт, які викликають труднощі (за потреби).
Аудиторна контрольна робота	Електронне тестування	Не використовуються.
Екзамен	Усний екзамен	<u>Використовуються активно:</u> для виконання студентами практичних завдань екзаменаційного білета.

Одне з питань, що розглядається у межах лабораторної роботи №10, – одержання відомостей про пристрої, зокрема про розділи жорсткого диску, шляхом виведення вмісту каталогу `/dev/disk/by-id` у розширеному форматі (рис. 3.13).

```
student@lab:~$ ls -l /dev/disk/by-id
загалом 0
lrwxrwxrwx 1 root root 9 лют 18 14:11 ata-VBOX_CD-ROM_VB2-01700376 -> ../../sr0
lrwxrwxrwx 1 root root 9 лют 18 14:11 ata-VBOX_HARDDISK_VB34c06378-1d9aeaca ->
../../sda
lrwxrwxrwx 1 root root 10 лют 18 14:11 ata-VBOX_HARDDISK_VB34c06378-1d9aeaca-par
t1 -> ../../sda1
lrwxrwxrwx 1 root root 10 лют 18 14:11 ata-VBOX_HARDDISK_VB34c06378-1d9aeaca-par
t2 -> ../../sda2
lrwxrwxrwx 1 root root 10 лют 18 14:11 ata-VBOX_HARDDISK_VB34c06378-1d9aeaca-par
t5 -> ../../sda5
```

Рис. 3.13. Методичний посібник «Операційні системи та системне програмування» для бакалаврів інформатики:
приклад вмісту каталогу `/dev/disk/by-id`

У випадку, зображеному на рис. 3.13, до комп'ютера (віртуальної машини на базі VirtualBox) підключено віртуальний оптичний привід (файл sr0) та віртуальний жорсткий диск (весь диск – файл sda, розділи – файли sda1, sda2, sda5). Однак залежно від апаратно-програмної платформи, на якій виконується лабораторна робота, ці відомості можуть суттєво різнитися, що враховано в інструктивно-методичних матеріалах.

Передусім, виконуючи завдання на віртуальній машині у комп'ютерному класі, студент, ймовірно, не матиме змоги побачити у виводі флеш-носіїв, SD-карт тощо. Для цього VirtualBox потребує встановлення відповідного віртуального носія, для чого, у свою чергу, необхідні права адміністратора. З метою демонстрації ситуації з під'єднанням подібних носіїв всім студентам, незалежно від використовуваних ними програмних та апаратних засобів, у методичному посібнику розглянуто приклад, у якому ці носії підключено, і їм відповідають файли sdb*, sdc*, sdd* і т.п. Окремо описано також випадки, коли підключено жорсткі диски з IDE-інтерфейсом (таким пристроям відповідають файли hda*), коли використовуються віртуальні диски формату .xvd (Xen Virtual Disk – віртуальний диск Xen, їм відповідають файли xvd*, причому саме даний формат часто застосовується для інстансів Amazon EC2), оптичні приводи тощо.

Дещо подібною є ситуація з монтуванням флеш-носія. Для студентів, які мають змогу здійснити підключення флеш-носія на практиці (приміром, якщо ці студенти використовують власні ноутбуки), подано відповідні практичні завдання, супроводжувані інструкціями і поясненнями, причому як щодо автоматичного монтування, так і для монтування вручну. Для решти студентів подано детально проілюстрований опис процесу монтування, а також згаданий вище приклад вмісту каталогу /dev/disk/by-id, коли ці носії вже змонтовано.

Важливою особливістю лабораторних робіт з основ системного програмування у віртуалізованій ОС є потреба обміну файлами з цією ОС. У деяких випадках ця потреба відсутня (коли студент виконує і здає лабораторні роботи на власному ноутбучі; коли студент працює з хмарним сервісом через

власний обліковий запис). Однак у решті випадків необхідно організувати обмін файлами між основною і віртуалізованою ОС, причому для різних засобів механізми такого обміну не співпадають.

Методичним посібником «Операційні системи та системне програмування» [44] передбачено інструкції для обміну файлами з кожним засобом із комбінації засобів віртуалізації (VirtualBox, Amazon EC2) з урахуванням найпоширеніших частинних випадків.

Наприклад, у самотійній роботі №1 («Основи роботи з VirtualBox») для обміну з віртуальною машиною на основі VirtualBox запропоновано механізм спільних каталогів (shared folders), причому з окремим розглядом випадку, коли віртуалізована ОС має графічний інтерфейс, і випадку, коли у віртуалізованій ОС встановлено лише інтерфейс командного рядка. Враховується потреба встановлення у VirtualBox гостьових доповнень для уможливлення роботи зі спільними каталогами. Процедура інсталювання гостьових доповнень різниться для графічного інтерфейсу та інтерфейсу командного рядка, тому самотійна робота №1 описує обидва випадки.

У самотійній роботі №2 («Основи роботи з Amazon EC2») для обміну файлами інстансом Amazon EC2 з ОС Linux та основною ОС Windows (або ОС Linux з графічним інтерфесом) пропонується використання FTP-клієнта FileZilla, й водночас надається посилання на опис такого обміну через команду scp (якщо основна ОС – Linux, незалежно від інтерфейсу). Для інстансів з Windows пропонується вирішити проблему шляхом налаштувань відповідного rdp-файлу (один з розділів жорсткого диску комп'ютера користувача вказується як локальний ресурс для інстансу).

Поради щодо вибору засобів для виконання лабораторних робіт з основ системного програмування наводяться на початку лабораторної роботи №7 «Програмування процесів і потоків у Linux», однак вони стосуються і лабораторної роботи №12 «Програмування роботи з файлами у Linux» (рис. 3.14).

Таблиця 2.8.1. Порівняння запропонованих наборів інструментів для вивчення основ програмування під Linux за деякими факторами				
Фактори \ Набори інструментів	а) VirtualBox / Xubuntu 14.04 / GUI / g++ / 1 Гб RAM	б) VirtualBox / Xubuntu 14.04 / GUI / g++ / Code::Blocks / 1 Гб RAM	в) VirtualBox / Ubuntu 14.04 / CLI / g++	г) Amazon EC2 / Ubuntu 14.04 / CLI / g++ / Інтернет
На комп'ютері, за яким ви працюєте, менше 1 Гб оперативної пам'яті?	-	-	+	+
Програмуєте здебільшого там, де немає з'єднання з Інтернетом?	+	+	+	-
Дуже невпевнено почуваєтеся без графічного інтерфейсу?	+	+	-	-
Бажаєте неодмінно працювати з IDE?	-*	+	-**	-**

Позначка "+" означає, що відповідний набір інструментів підходить в описаній ситуації.

Позначка "-" свідчить, що відповідний набір інструментів не підходить в описаній ситуації.

Примітки

* Хоча в наборі інструментів **а** відсутнє IDE Code::Blocks, за даної конфігурації його можна довантажити (див. *самостійна робота №3*). Тоді набір інструментів **а**, по суті, зведеться до набору інструментів **б**.

** Заради справедливості треба зазначити, що у системі, в якій доступний лише інтерфейс командного рядка (CLI), можна встановити редактор Emacs, про який уже йшлося вище (наразі не описаний у даному посібнику). Emacs не є IDE (принаймні, у звичному для більшості користувачів розумінні), це радше розширений текстовий редактор, робота у якому потребує попереднього навчання, однак можливості Emacs наближають його до функціоналу IDE.

*Рис. 3.14. Методичний посібник «Операційні системи та системне програмування» для бакалаврів інформатики:
фрагмент рекомендацій для студентів щодо вибору ПЗ
для лабораторних робіт з основ системного програмування*

Зазвичай перед початком виконання першої із лабораторних робіт, які передбачали програмування у Linux, студенти одержували рекомендацію використовувати набори інструментів (в) чи (а) як одні з найменш вимогливих до системних ресурсів. Набір інструментів (б) обирали студенти, які висловлювали бажання працювати з інтегрованим середовищем розробки і мали для цього відповідні апаратні ресурси (наприклад, Linux з графічним

інтерфейсом на власному ноутбучі), а варіант (з) орієнтований на студентів з власним обліковим записом в Amazon Web Services, а також епізодично використовувався для створення програмного коду, коли інші інструменти були недоступні. Втім, варто зазначити, що незалежно від обраного студентом ПЗ, однією з вимог до захисту лабораторних робіт з основ системного програмування є вміння компілювати програмний код засобами командного рядка.

Окрім рекомендацій щодо вибору ПЗ для виконання робіт із основ системного програмування, методичний посібник містить також розгляд процедури встановлення більшості цих засобів (компілятор g++ у лабораторній роботі №7, IDE Code::Blocks у самостійній роботі №3), а також опис деяких проблем, які можуть виникати під час інсталювання, та ідей для їх вирішення.

Відповідно до варіативного підходу, перелік наборів програмних та апаратних засобів, запропонованих у лабораторній роботі №7 і на рис. 3.14, може бути розширено. Так, у 2017-18 н. р. під час проведення лабораторних робіт з дисципліни «Операційні системи та системне програмування», які передбачали навчання основ системного програмування, було застосовано хмарне IDE Cloud9. Модифіковану комбінацію засобів віртуалізації із доданим середовищем розробки Cloud9 показано на рис. 3.15.

	Linux	Windows
Адміністрування	VirtualBox Amazon EC2 Cloud9 ☒ без віртуалізації	Amazon EC2 ☒ без віртуалізації
Програмування	Cloud9 VirtualBox ± Code::Blocks Amazon EC2 ☒ без віртуалізації	☒ без віртуалізації + Code::Blocks Amazon EC2 + Code::Blocks

Рис. 3.15. Комбінація засобів віртуалізації для навчання ОС бакалаврів інформатики із доданим хмарним IDE Cloud9

Визначення ефекту від використання хмарних інтегрованих середовищ розробки, зокрема IDE Cloud9, виходить за хронологічні межі психолого-педагогічного експерименту, проведеного відповідно до даного дослідження. Вважаємо за доцільне подальше вивчення цього питання.

Засоби віртуалізації використовуються також під час консультування та захисту лабораторних робіт. Перевагою варіативного підходу у такому разі є змога проводити ці види роботи в різних аудиторіях і на різних пристроях. Наприклад, один чи декілька інстансів Amazon EC2 з ОС Linux можуть бути швидко запуснені з будь-якого комп'ютера, що має доступ до Інтернету. Це дозволяє викладачеві оперативно реагувати на запитання студентів та стимулює студентів до активної роботи.

Висновки до розділу 3

У розділі 3 розроблено та описано методику застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики.

Мета застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики полягає у створенні умов для формування в студентів інформатичних компетентностей щодо ОС. Відповідно до мети, визначено *завдання* застосування Unix-подібних ОС: (1) навчання основ теорії ОС; (2) освоєння основ адміністрування ОС; (3) навчання основ системного програмування; (4) створення передумов для подальшої самоосвіти з ОС.

Кожному завданню відповідають інформатичні компетентності бакалаврів інформатики щодо ОС, визначені у розділі 1, та очікувані результати навчання. Разом зазначені мета, завдання, компетентності й результати навчання утворюють чотирирівневу структуру.

Обґрунтовано необхідність індивідуального добору засобів віртуалізації Unix-подібних ОС, потребу залучення альтернативних засобів віртуалізації поряд з основними.

Застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики доцільно здійснювати згідно з варіативним підходом.

Варіативний підхід до застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики полягає у поєднанні кількох засобів віртуалізації та, можливо, варіантів навчання ОС без використання віртуалізації задля відповідності індивідуальним особливостям проведення курсу, індивідуальним потребам студентів і для забезпечення стійкості у разі програмних та апаратних збоїв. Підхід передбачає наступні завдання: (1) ознайомити викладачів та студентів з різноманітністю засобів віртуалізації; (2) навести приклад поєднання кількох засобів віртуалізації у межах одного курсу з ОС; (3) здійснити адаптування навчально-методичних матеріалів до використання різних засобів віртуалізації.

Розроблена методика застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики містить мету, завдання,

форми, методи і засоби. Організацію навчального процесу відповідно до пропонованої методики варто здійснювати за методичним посібником з дисципліни «Операційні системи та системне програмування» для навчання майбутніх бакалаврів інформатики та методичними рекомендаціями «Технології віртуалізації у навчанні операційних систем бакалаврів інформатики» для викладачів закладів вищої освіти.

Розглянуто *зміст, форми, методи і засоби навчання під час застосування технологій віртуалізації Unix-подібних ОС на прикладі змістового модуля «Файлові системи»*. З метою вибору змістового модуля для детального розгляду проведено експертне опитування. До опитування залучено 15 респондентів – викладачів ЗВО України, яким було запропоновано оцінити важливість змістових модулів курсу з ОС для бакалаврів інформатики. Констатовано невисоку однотайність експертів (коефіцієнт узгодженості Кендалла $W=0,33$), визначено її можливі причини, виокремлено групу змістових модулів, котрі одержали найвищі суми оцінок, серед яких – змістовий модуль «Файлові системи».

На прикладі змістового модуля «Файлові системи» описано зміст навчання (лекційна частина курсу, лабораторні та самостійні роботи), форми організації навчання (лекція-презентація, лекція-бесіда, лабораторна робота, аудиторна контрольна робота, екзамен), методи навчання (розповідь, пояснення, бесіда, інструктаж, пояснення; демонстрація, спостереження; лабораторна робота та ін.), засоби навчання (апаратне і програмне забезпечення, зокрема засоби віртуалізації; підручники і посібники; освітні і професійні веб-ресурси; звичайні та маркерні дошки, роздаткові матеріали тощо). Розглянуто використання засобів віртуалізації у поєднанні з формами та методами навчання згідно з пропонованою методикою.

Основні результати дослідження, проведеного у межах розділу 3, відображено у публікаціях [39; 43; 44; 45; 47; 50; 51; 148; 149; 160; 161].

РОЗДІЛ 4.

ОРГАНІЗАЦІЯ, ПРОВЕДЕННЯ ТА РЕЗУЛЬТАТИ ПЕДАГОГІЧНОГО ЕКСПЕРИМЕНТУ

4.1. Організація педагогічного експерименту

За С. У. Гончаренком, психолого-педагогічний експеримент є комплексним методом дослідження, що «забезпечує науково-об'єктивну і доказову перевірку правильності обґрунтованої на початку дослідження гіпотези» [52, с. 253].

Під час психолого-педагогічного експерименту здійснюється «активне формування психічних якостей, процесів, що досліджуються безпосередньо під час навчально-виховної діяльності» [84].

У межах даного дослідження експеримент передбачав спостереження за процесом застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики. Крім того, формувальний етап експерименту був спрямований на перевірку сформульованої раніше *гіпотези* про покращення процесу формування інформатичних компетентностей щодо ОС у бакалаврів інформатики у разі, якщо навчання Unix-подібних операційних систем здійснюватиметься на основі розробленої методики застосування технологій віртуалізації.

Педагогічний експеримент було організовано як природний (а саме психолого-педагогічний) порівняльний експеримент. Відповідно, спостереження досліджуваного процесу виконувалося в умовах, наближених до умов, звичних для піддослідних (природний експеримент), а спеціальні педагогічні зміни здійснювалися в експериментальній групі й не здійснювалися у контрольній (порівняльний експеримент) [52, с. 254].

Організація і проведення педагогічного експерименту відбувалася у наступні етапи.

- 1) Констатувальний етап (2013-2015).
- 2) Формувальний етап (2015-2018).

На констатувальному етапі експерименту (2013-2015) було поставлено за мету дослідити процес навчання ОС у вітчизняних та закордонних ЗВО, включаючи особливості застосування для навчання ОС технологій віртуалізації Unix-подібних ОС, та з'ясувати проблеми, які супроводжують застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики. Відповідно до зазначеної мети, протягом констатувального етапу реалізовано наступні заходи.

- Проаналізовано вітчизняний та закордонний досвід навчання ОС і роль у ньому засобів віртуалізації (шляхом опрацювання науково-методичних публікацій, підручників, навчальних програм, вітчизняних галузевих стандартів та закордонних рекомендацій щодо розробки навчальних програм, матеріалів з відкритих освітніх ресурсів низки ЗВО світу).

- Проведено бесіди зі студентами, які вивчали курс з ОС, вітчизняними викладачами відповідного курсу та фахівцями у галузі.

- Проаналізовано власний досвід викладання курсу з ОС бакалаврам інформатики у Житомирському державному університеті імені Івана Франка.

- Здійснено аналіз підсумкових балів, одержаних майбутніми бакалаврами інформатики після вивчення курсу з ОС, а також аналіз результатів проходження студентами контрольного тестування «Unix-подібні операційні системи».

- Проведено анкетування викладачів щодо їхнього досвіду застосування технологій віртуалізації під час викладання курсу з ОС.

- Розроблено методику застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики та підготовлено відповідне інструктивно-методичне забезпечення.

- Уточнено гіпотезу дослідження.

- Здійснено підбір експериментальної бази для формувального етапу експерименту. Сформовано контрольну групу (КГ) та експериментальну групу (ЕГ), перевірено однорідність вибірок на основі критерію типу Колмогорова-Смирнова та кутового перетворення Фішера.

Метою *формульованого етапу експерименту* (2015-2018) була перевірка раніше сформульованої та уточненої гіпотези про покращення процесу формування інформатичних компетентностей бакалаврів інформатики щодо ОС у разі, якщо навчання Unix-подібних ОС здійснюватиметься на основі розробленої методики застосування технологій віртуалізації.

Задля досягнення поставленої мети реалізовано наступні заходи.

- Реалізовано навчання курсу з ОС відповідно до розробленої методики застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики в експериментальній групі.

- Організовано проведення контрольних заходів у КГ та ЕГ та одержано результати навчання піддослідних студентів.

- Проаналізовано одержані експериментальні дані (критерій типу Колмогорова-Смирнова та кутове перетворення Фішера) з метою виявлення статистично значущих відмінностей між рівнями навчальних досягнень у КГ та ЕГ.

- Узагальнено результати аналізу експериментальних даних й сформульовано висновки щодо ефективності розробленої методики і можливостей її подальшого використання.

- Розроблено методичні рекомендації із застосування технологій віртуалізації у навчання ОС бакалаврів інформатики. Уточнено процедуру добору засобів віртуалізації для навчання операційних систем бакалаврів інформатики.

- Проведено експертне опитування з метою вибору змістового модуля для опису на його прикладі розробленої методики, проаналізовано результати опитування.

- Впроваджено результати дослідження у процес підготовки бакалаврів інформатики у вітчизняних ЗВО.

4.2. Проведення та аналіз результатів констатувального етапу експерименту

Метою констатувального етапу експерименту було дослідження процесу навчання ОС у вітчизняних та закордонних ВНЗ, включаючи особливості застосування для навчання ОС технологій віртуалізації Unix-подібних ОС, та

з'ясування проблем, які супроводжують застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики.

На початку констатувального етапу експерименту здійснено *аналіз вітчизняного та закордонного досвіду навчання ОС*. Особливу увагу приділено залученню у процесі навчання ОС засобів віртуалізації, передусім – засобів віртуалізації Unix-подібних ОС. Вітчизняний досвід навчання ОС (п. 1.2.2) досліджувався шляхом опрацювання науково-методичних публікацій, підручників, навчальних програм, галузевого стандарту вищої освіти України (на момент проведення констатувального етапу експерименту діяв стандарт [98; 99], згодом було додатково опрацьовано і враховано проект нового галузевого стандарту [100]). Ознайомлення із закордонним досвідом навчання ОС здійснювалося за науково-методичними публікаціями, підручниками, навчальними програмами, керівними принципами розробки навчальних програм Computer Science Curricula 2013 [213], а також матеріалів з веб-ресурсів низки ЗВО світу. Опрацьовано передусім досвід ЗВО, які є лідерами у рейтингах за підготовкою зі спеціальності «Computer Science», а також досвід низки закладів освіти з дещо нижчими аналогічними рейтингами, якщо такий досвід було досить детально опубліковано у той чи інший спосіб. Закордонний досвід навчання ОС розглянуто у п. 1.3.

У результаті аналізу вітчизняного та закордонного досвіду одержано основу для розробки переліку науково-предметних та професійно-практичних компетентностей, які підлягають формуванню в бакалаврів інформатики у межах курсу з ОС, відповідних результатів навчання, критеріїв та показників їх оцінювання. Також було відзначено зокрема наступні особливості курсу з ОС.

- У проаналізованих вітчизняних ЗВО навчання ОС суттєво відрізнялося за обсягом відведених годин та навчальними програмами.
- У більшості проаналізованих випадків технології віртуалізації використовувалися під час виконання лабораторних робіт, а у частині випадків (передусім у закордонних ЗВО) розглядалися у межах лекційної частини курсу.
- У закордонних ЗВО під час виконання лабораторних робіт

застосовувався один або декілька засобів віртуалізації. Відзначено суттєву різноманітність конкретних використовуваних засобів віртуалізації.

- Вивчення Unix-подібних ОС є пріоритетним у переважній більшості проаналізованих випадків загалом і в абсолютній більшості випадків, що стосувалися закордонних ЗВО.

- У практичній частині курсу може у тому чи іншому співвідношенні поєднуватися навчання системного програмування та основ адміністрування ОС. У закордонних курсах з ОС перевага належить системному програмуванню, у вітчизняних – системному програмуванню у курсах для інженерних спеціальностей та адмініструванню – для педагогічних.

- Закордонними курсами передбачено навчання основ роботи із застосуванням ПЗ, у тому числі віртуалізаційним, за відповідними інструктивно-методичними матеріалами (розробленими для курсу чи універсальними).

Також на констатувальному етапі експерименту проведено *бесіди* зі студентами, які вивчають курс з ОС, викладачами курсу з ОС, фахівцями у галузі. Було *проаналізовано власний досвід* викладання курсу з ОС бакалаврам інформатики у Житомирському державному університеті імені Івана Франка (дисципліна «Операційні системи», згодом – «Операційні системи та системне програмування»).

Було відзначено зокрема наступні проблеми, пов'язані із застосуванням технологій віртуалізації під час навчання Unix-подібних ОС.

- Апаратні та програмні збої.
- Низька швидкість роботи віртуалізованої ОС.
- Відсутність адаптованих інструкцій до віртуалізаційного ПЗ.
- Прив'язаність лабораторних занять до конкретних аудиторій.
- Неоднаковість умов, у яких здійснюється самостійна робота студентів з віртуалізованою ОС (місце, потужність апаратної складової, наявність Інтернет-з'єднання, основна ОС тощо).

Крім того, у ході проведення бесід відзначено, що коли курс передбачає навчання і Unix-подібної ОС (наприклад, Linux), і ОС Windows, то залучені до бесід

викладачі та студенти часто характеризували рівень навчальних досягнень, пов'язаних з Unix-подібними ОС, як загалом нижчий, порівняно з рівнем навчальних досягнень з курсу в цілому, у тому числі порівняно з рівнем навчальних досягнень, пов'язаних з ОС Windows. Аналогічні прояви було зафіксовано й на основі власного досвіду викладання курсу з ОС. Констатовано необхідність цілеспрямованої експериментальної перевірки існування такої тенденції.

З метою експериментальної перевірки припущення про загалом нижчий рівень навчальних досягнень, пов'язаних з Unix-подібними ОС, порівняно з рівнем навчальних досягнень з усього курсу з ОС, було здійснено *аналіз балів майбутніх бакалаврів інформатики* (на момент проведення дослідження – напрям підготовки 6.040302 «Інформатика») Житомирського державного університету імені Івана Франка (ЖДУ імені Івана Франка) та Дрогобицького державного педагогічного університету імені Івана Франка (ДДПУ імені Івана Франка), загалом 98 осіб (табл. 4.1).

Таблиця 4.1

Студенти, які брали участь у констатувальному етапі педагогічного експерименту

	<i>Житомирський державний університет імені Івана Франка</i>			<i>Дрогобицький державний педагогічний університет імені Івана Франка</i>		
Навчальний рік, у якому вивчалася дисципліна	2010-11	2011-12	2013-14	2011-12, 2012-13	2012-13, 2013-14	2013-14
Курс, на якому вивчалася дисципліна	II	II	II	III і IV	III і IV	III
Кількість студентів у групі	26	22	22	9	3	16

Було проаналізовано наступні бали студентів:

- підсумковий бал з дисципліни «Операційні системи та системне програмування»;
- середній бал за лабораторні роботи, присвячені Unix-подібним ОС;
- бал, набраний під час проходження тестового опитування «Unix-подібні операційні системи».

Важливо відзначити, що у проаналізованих випадках курс з ОС

передбачав поряд із навчанням Unix-подібної ОС (Linux), також навчання ОС Windows і розгляд питань з теорії ОС загалом. Підсумковий бал враховував усі відповідні навчальні досягнення студентів, а тому у низці випадків нижчий рівень навчальних досягнень, пов'язаних з Linux, міг компенсовуватися вищим рівнем інших навчальних досягнень (робота з ОС Windows; загальні теоретичні питання), за рахунок чого підсумковий бал виходив би вищим.

Для студентських груп ЖДУ імені Івана Франка 2010-11 і 2011-12 н. р. порівняння підсумкового балу з дисципліни «Операційні системи та системне програмування» та середнього балу за лабораторні роботи, присвячені Linux (рис. 4.1 і рис. 4.2 відповідно), показало, що середні бали за лабораторні роботи

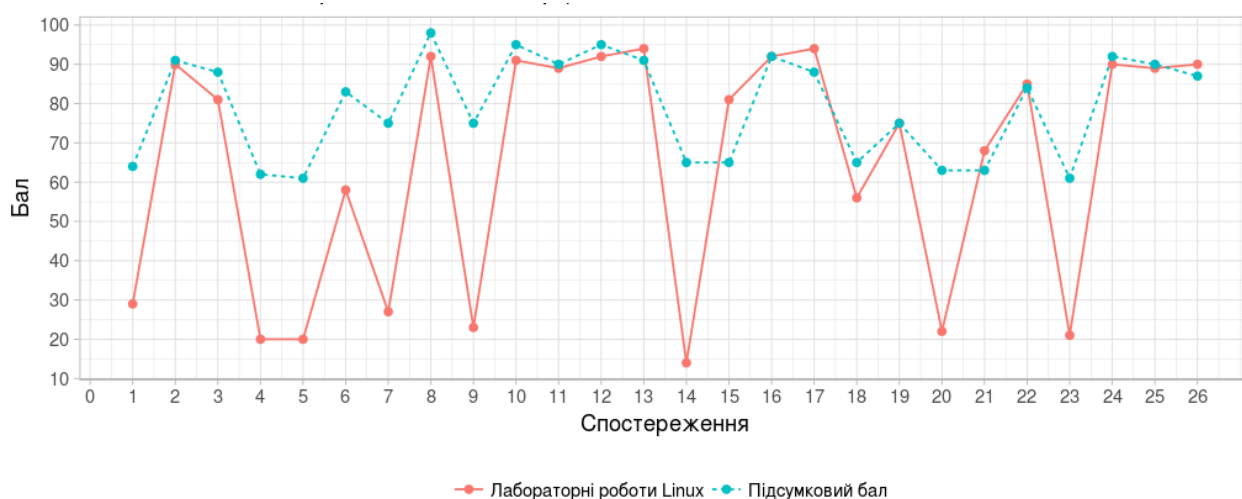


Рис. 4.1. Порівняння балів студентів з дисципліни «Операційні системи та системне програмування» (ЖДУ імені Івана Франка, 2010-11 н. р.)

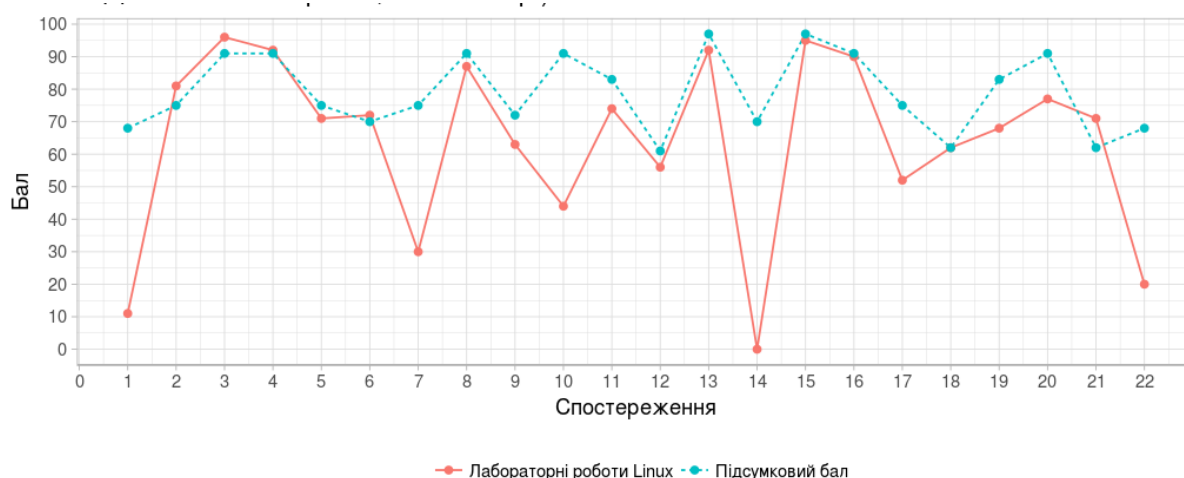


Рис. 4.2. Порівняння балів студентів з дисципліни «Операційні системи та системне програмування» (ЖДУ імені Івана Франка, 2011-12 н. р.)

з Linux загалом нижчі за підсумкові бали з дисципліни. Поодинокі випадки, коли мала місце протилежна ситуація (середній бал за лабораторні роботи з Linux, навпаки, дещо перевищував підсумковий бал з дисципліни) передусім стосувалися студентів, які виявляли підвищений інтерес до вивчення саме ОС Linux і/або мали певний досвід роботи з нею до початку проходження курсу.

Для студентів ЖДУ імені Івана Франка, для яких навчання дисципліни «Операційні системи та системне програмування» припало безпосередньо на період проведення констатувального етапу експерименту (студенти, які вивчали дисципліну у 2013-14 н. р.), наприкінці вивчення курсу було також проведено тестове опитування «Unix-подібні операційні системи». Опитування містило питання теоретичного і практичного характеру, присвячені саме Unix-подібним ОС, і проводилося у формі тестового опитування як бланкове та електронне тестування (додаток П). Тест містив дихотомічні запитання, запитання зі стандартизованою низкою відповідей та відкриті короткі питання. Порівняння підсумкового балу з дисципліни «Операційні системи та системне програмування», середнього балу за лабораторні роботи, присвячені Linux, та балу за контрольне опитування «Unix-подібні операційні системи» (рис. 4.3) також показало, що середні бали за лабораторні роботи з Linux переважно нижчі за підсумкові бали з дисципліни. Бали, одержані за тестове опитування

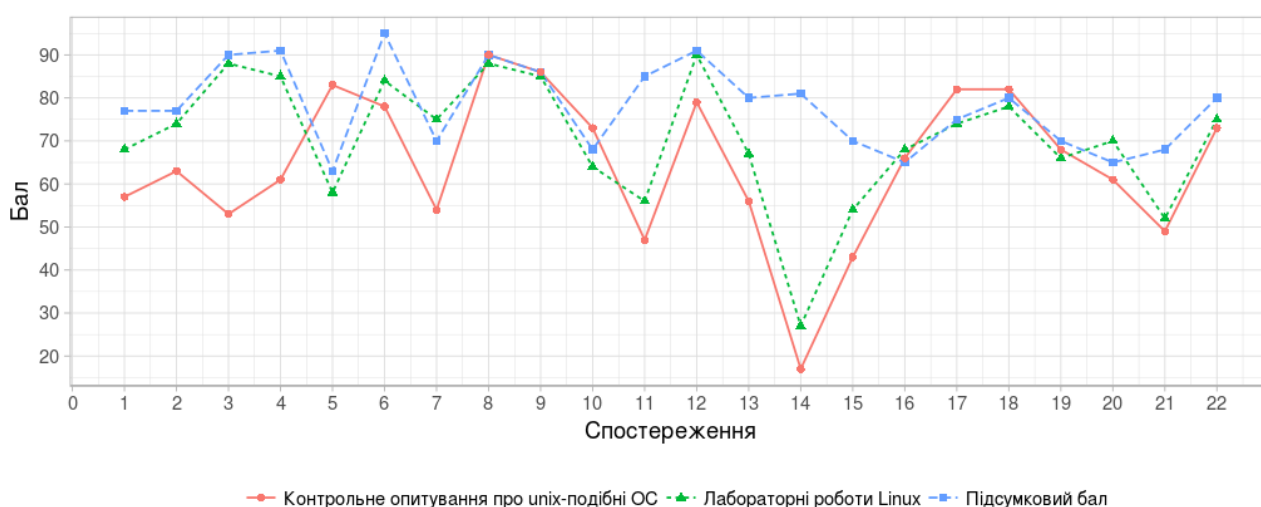


Рис. 4.3. Порівняння балів студентів з дисципліни «Операційні системи та системне програмування» (ЖДУ імені Івана Франка, 2013-14 н. р.)

«Unix-подібні операційні системи», також переважно нижчі за підсумковий бал і лише у поодиноких випадках співпадають з ним або перевищують його.

Задля перевірки припущення про загалом нижчий рівень навчальних досягнень, пов'язаних з Unix-подібними ОС, порівняно з загальним рівнем навчальних досягнень з ОС, в умовах іншого ЗВО було також проаналізовано бали, одержані майбутніми бакалаврами інформатики ДДПУ імені Івана Франка (див. табл. 4.1). Порівнювався підсумковий бал з дисципліни «Операційні системи та системне програмування» та середній бал за лабораторні роботи, присвячені Unix-подібним ОС (рис. 4.4, 4.5).

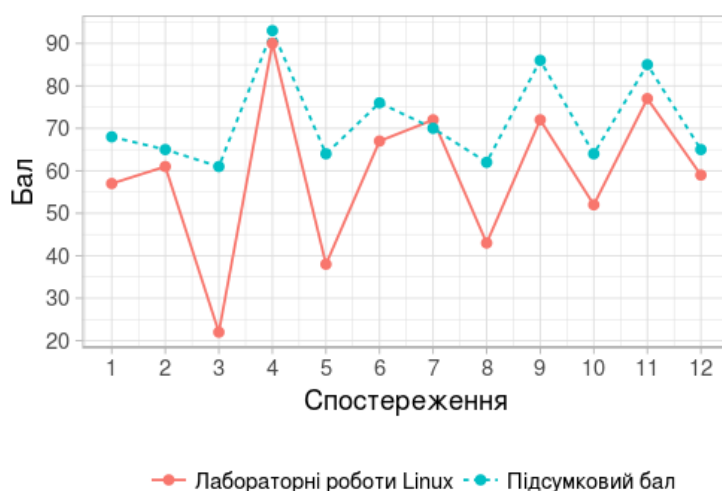


Рис. 4.4. Порівняння балів студентів з дисципліни «Операційні системи та системне програмування» (ДДПУ імені Івана Франка, II семестр 2011-12 н. р. – I семестр 2012-13 н. р.; II семестр 2012-13 н. р. – I семестр 2013-14 н. р.)

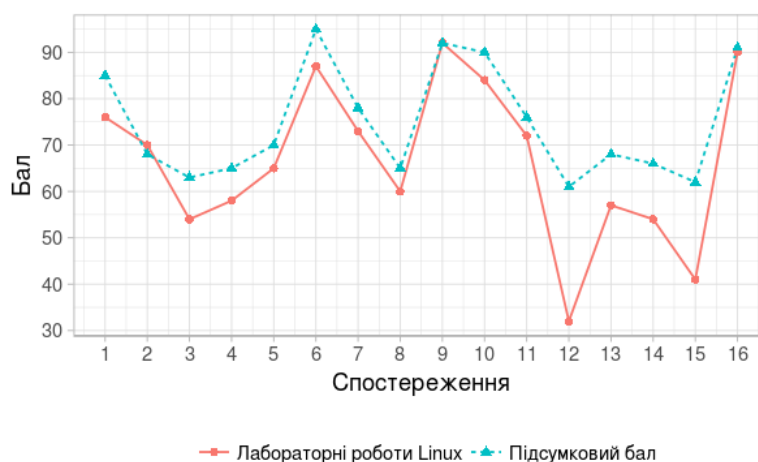


Рис. 4.5. Порівняння балів студентів з дисципліни «Операційні системи та системне програмування» (ДДПУ імені Івана Франка, 2013-14 н. р.)

Як і у випадку з балами студентів ЖДУ імені Івана Франка, з діаграм на рис. 4.4 і рис. 4.5 видно, що середні бали за лабораторні роботи з ОС Linux загалом нижчі за підсумкові бали з дисципліни й лише у поодиноких випадках дорівнюють підсумковому балу або дещо перевищують його.

Така тенденція відповідає відзначеній раніше, у ході бесід з викладачами і студентами й на основі власного досвіду викладання, проблемі загалом нижчого рівня навчальних досягнень, пов'язаних з Unix-подібними ОС, порівняно з аналогічним рівнем для ОС Windows.

Задля детальнішого дослідження особливостей проведення курсу з ОС у вітчизняних ЗВО та ролі у цьому засобів віртуалізації проведено *опитування викладачів ЗВО України*, які викладали дисципліну «Операційні системи та системне програмування» та інші подібні дисципліни майбутнім бакалаврам інформатики, а також викладачів, які викладали аналогічні дисципліни студентам інших напрямів підготовки (спеціальностей), загалом 9 респондентів. Особливу увагу було приділено досвіду застосування опитуваними викладачами технологій віртуалізації, оцінювання ними важливості факторів вибору засобів віртуалізації та низки інших питань. Анкетування здійснювалося в електронній формі (додаток Р).

Враховуючи невеликий обсяг вибірки, основною метою аналізу результатів анкетування було не оцінювання відсоткового співвідношення проявів досліджуваних ознак, а виявлення відмінностей. Під час опрацювання даних анкет застосовано передусім візуальний аналіз.

Згідно з результатами опитування, назва дисципліни, у межах якої здійснювалося вивчення ОС, відрізнялося («Операційні системи та системне програмування», «Операційні системи», «Інформатика», «Інформаційні системи»). При цьому у межах відповідного курсу здебільшого розглядалися ОС Linux і ОС Windows, однак в окремих випадках дисципліна передбачала роботу лише з однією ОС – Linux, FreeBSD, Linux (рис. 4.6).

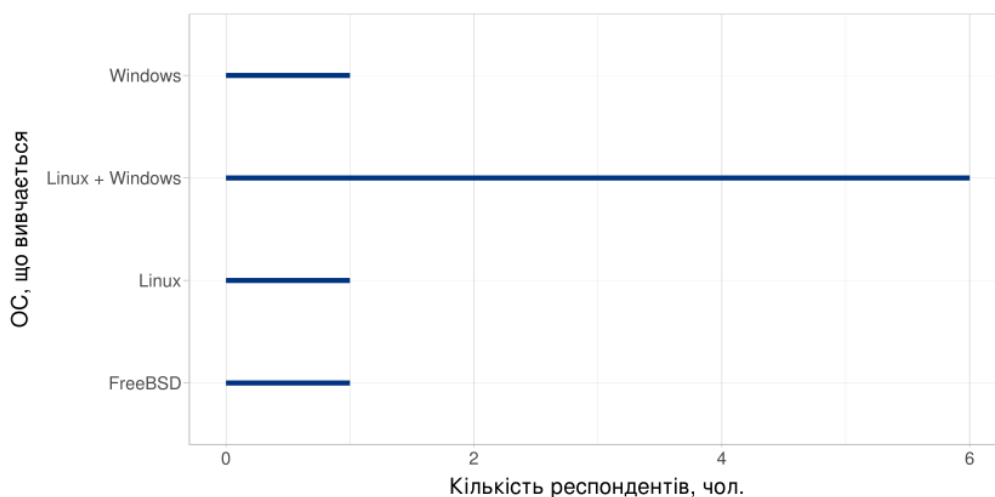


Рис. 4.6. Опитування викладачів курсу з ОС: розподіл відповідей респондентів за ОС, що вивчається

У випадках, коли здійснювалося вивчення Unix-подібних ОС, конкретні ОС чи їхні дистрибутиви також відрізнялися (рис. 4.7). Зафіксовано як випадки роботи з Debian-сумісними дистрибутивами Linux (Ubuntu, Mint, Debian), так і з RedHat-сумісними дистрибутивами Linux (Fedora), а також випадок, коли було передбачено роботу з ОС FreeBSD.

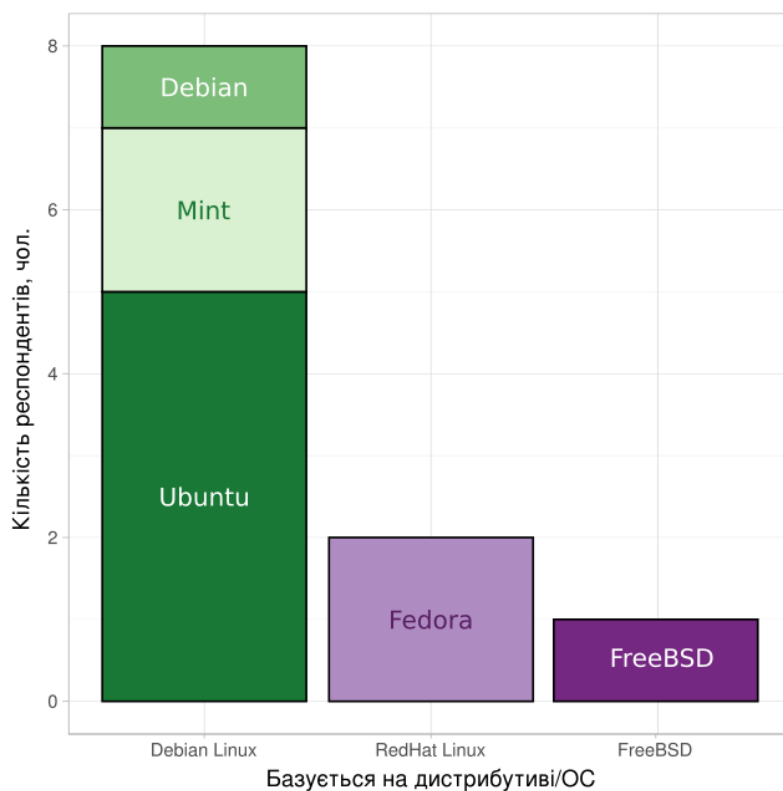


Рис. 4.7. Опитування викладачів курсу з ОС: розподіл відповідей респондентів за Unix-подібними ОС, у тому числі дистрибутивами Linux

Як видно з рис. 4.7, більшість респондентів вказали саме Debian-сумісні дистрибутиви, однак важливо зауважити, що у зв'язку з невеликим обсягом вибірки твердження про переважання саме таких дистрибутивів у програмах курсу з ОС потребує додаткового вивчення і нами не досліджувалося.

Також, згідно з даними анкет, на запитання про альтернативи засобам віртуалізації, використовувані під час навчання ОС (рис. 4.8), було зафіксовано відповіді про наявність окремого комп'ютерного класу з відповідною ОС і водночас – відповіді про відсутність такого класу й натомість застосування мультизавантаження. Мультизавантаження передбачає встановлення більш ніж однієї ОС на один комп'ютер. У разі, коли йдеться про мультизавантаження як про альтернативу засобам віртуалізації, може матися на увазі й організація

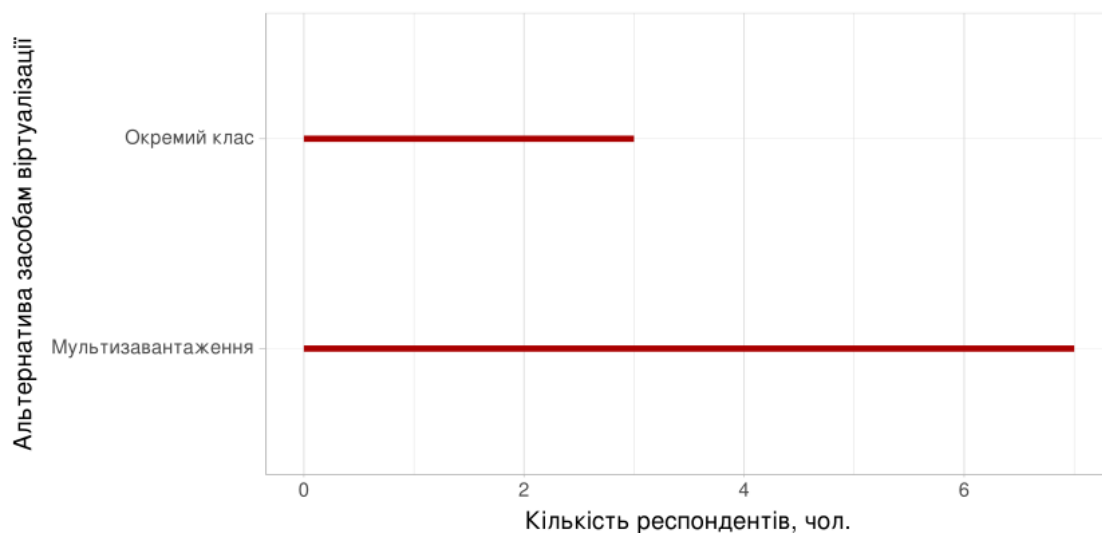


Рис. 4.8. Опитування викладачів курсу з ОС: розподіл відповідей респондентів за використовуваними альтернативами засобам віртуалізації

мультизавантаження у комп'ютерному класі, і встановлення другої ОС на власні комп'ютери окремих студентів (за бажанням студентів).

Відповіді на запитання про доступність викладачам апаратних засобів віртуалізації (рис. 4.9) засвідчили, що лише частина респондентів мали у своєму розпорядженні процесори з підтримкою відповідних технологій (локально, у комп'ютерному класі, чи віддалено, на сервері). Решта опитаних доступу до таких технологій у своєму закладі освіти не мали або не знали про це.

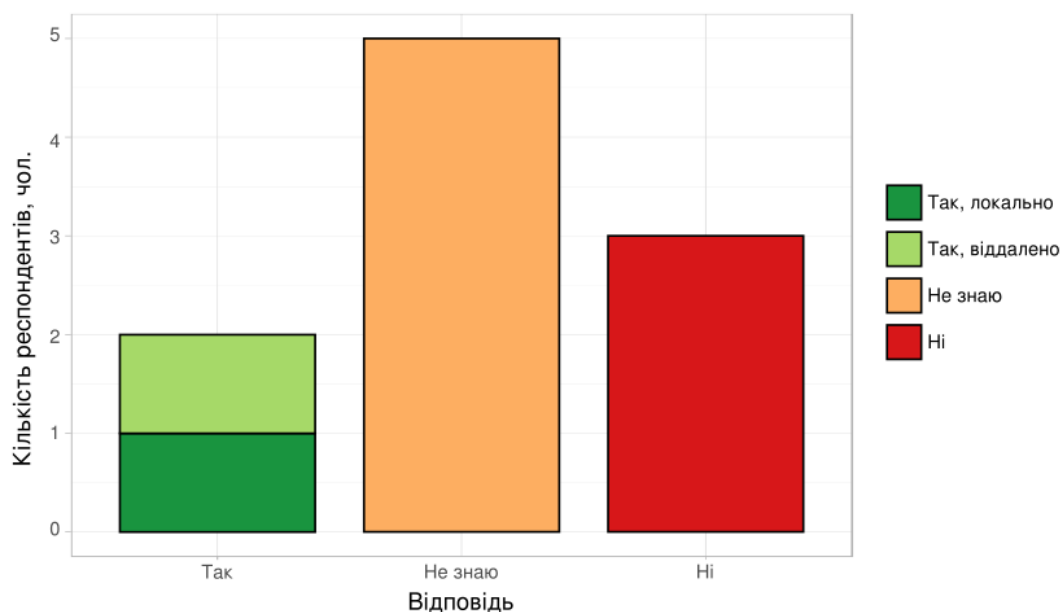


Рис. 4.9. Опитування викладачів курсу з ОС: розподіл відповідей респондентів за доступністю викладачам апаратних засобів віртуалізації

Різнилися й засоби віртуалізації, застосовувані опитаними викладачами на момент проведення анкетування (рис. 4.10), засоби віртуалізації, досвід роботи з якими мали респонденти (рис. 4.11), а також засоби віртуалізації, що їх респонденти розглядали як варіанти для використання у навчанні ОС у майбутньому (рис. 4.12).

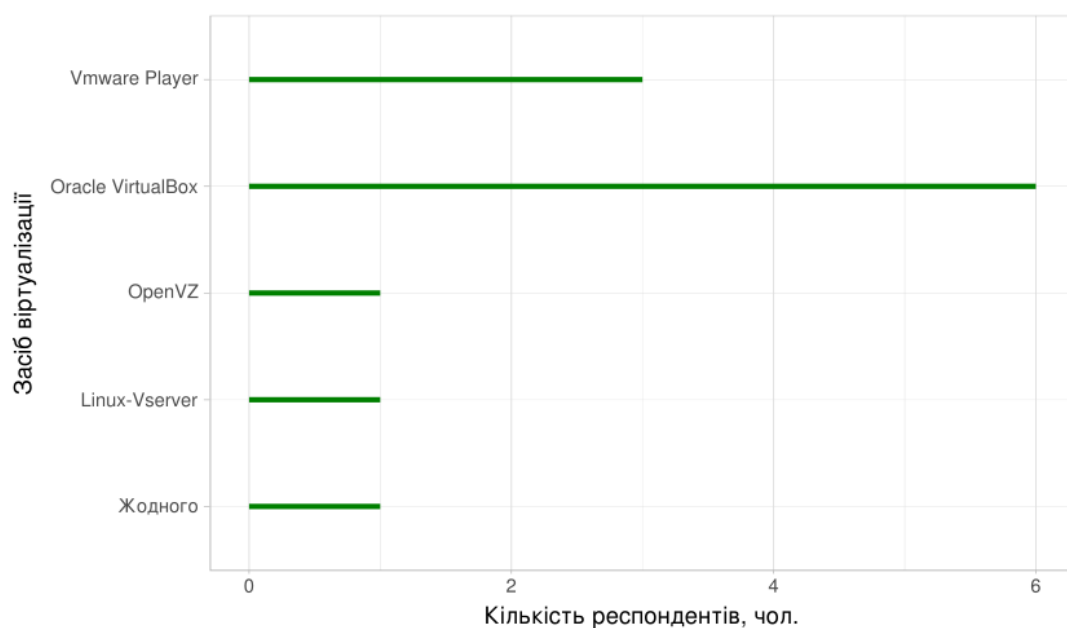


Рис. 4.10. Опитування викладачів курсу з ОС: розподіл відповідей респондентів за застосовуваними на момент опитування засобами віртуалізації

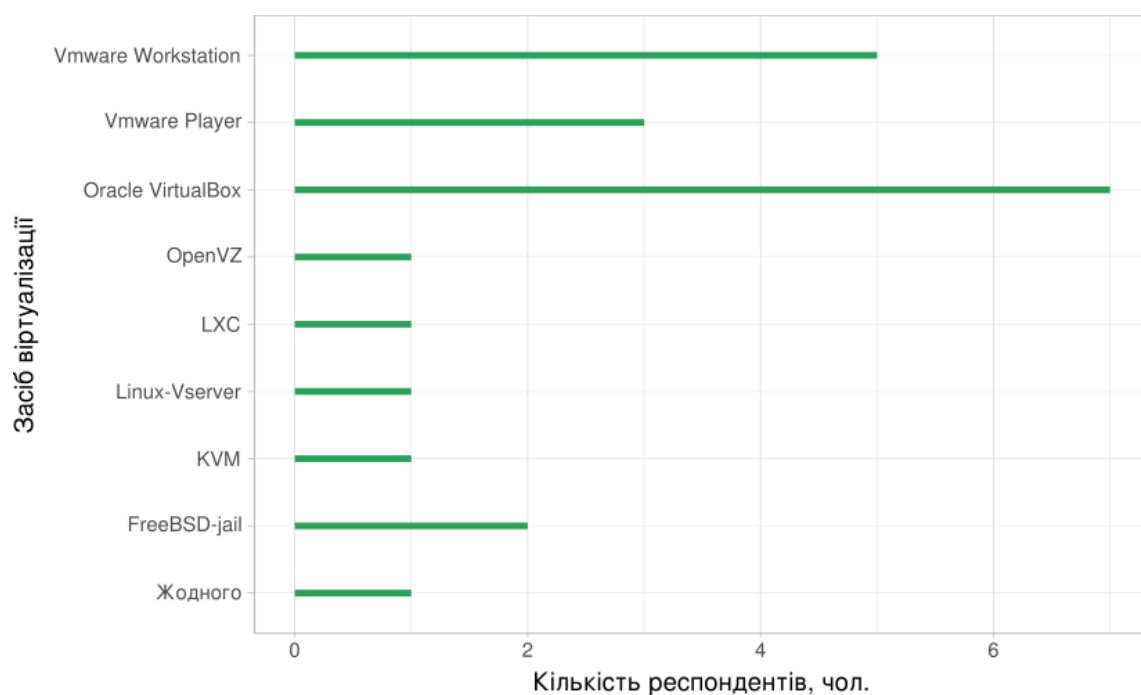


Рис. 4.11. Опитування викладачів курсу з ОС: розподіл відповідей респондентів за досвідом роботи із засобами віртуалізації

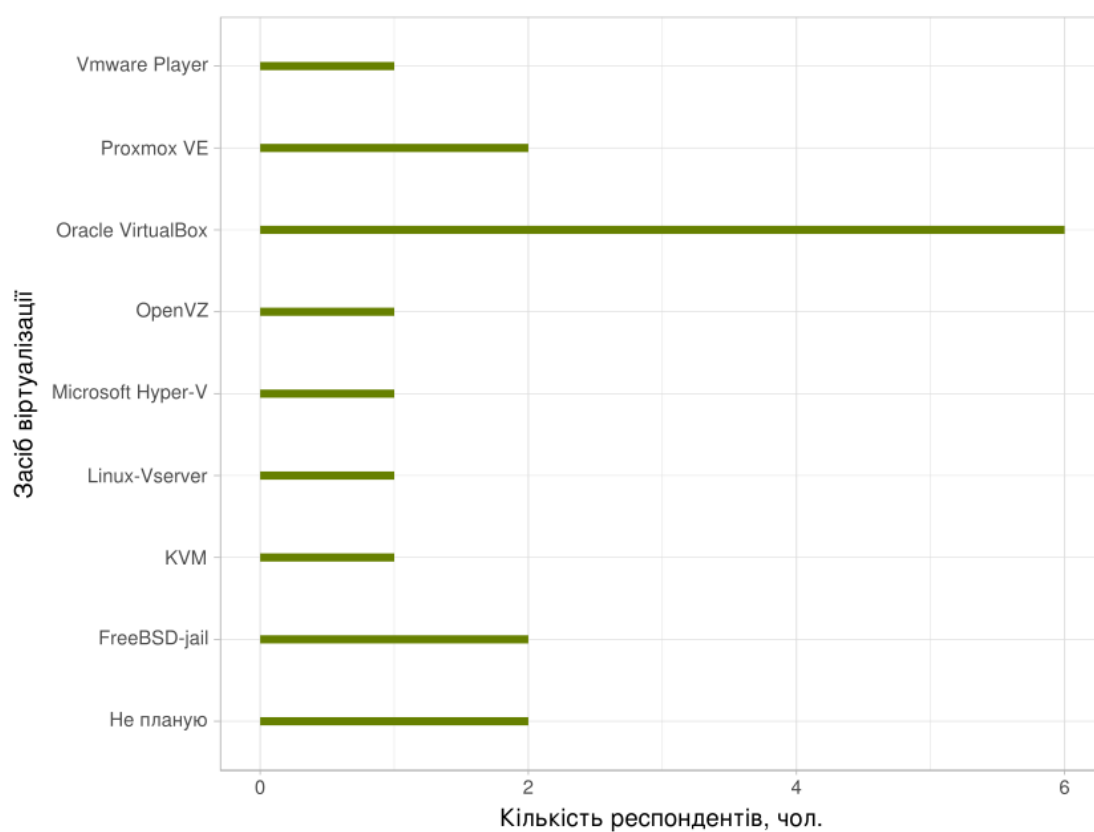


Рис. 4.12. Опитування викладачів курсу з ОС: розподіл відповідей респондентів за засобами віртуалізації, що їх респонденти розглядали як варіанти для застосування у курсі з ОС у майбутньому

Серед засобів віртуалізації, використовуваних респондентами на момент проведення опитування (рис. 4.10), викладачі вказували як гіпервізори II типу (Oracle VirtualBox, Vmware Player), так і віртуальні контейнери (Open VZ, Linux-VServer). Було також зафіксовано випадок, коли не застосовувався жодний засіб віртуалізації (натомість використовувалось мультизавантаження). Досвід роботи опитаних викладачів із засобами віртуалізації також був різноманітним (рис. 4.11): гіпервізори II типу (Vmware Workstation, Vmware Player, Oracle VirtualBox, KVM), віртуальні контейнери (Open VZ, Linux-VServer, FreeBSD-jail, LXC) або жодного. Серед засобів віртуалізації, що їх респонденти розглядали як варіанти для використання у навчанні ОС у майбутньому (рис. 4.12), було зазначено гіпервізори II типу (Vmware Player, Oracle VirtualBox, Microsoft Hyper-V, KVM), віртуальні контейнери (FreeBSD-jail, Linux-VServer, Open VZ), середовище віртуалізації Proxmox VE (поєднує віртуальні машини на базі KVM, віртуальні контейнери на базі Open VZ та веб-інтерфейс для віддаленого доступу). При цьому двоє опитаних викладачів, які на момент участі в опитуванні використовували засоби віртуалізації, вказали, що не планують застосовувати їх у майбутньому.

Також респондентам було запропоновано оцінити важливість факторів добору засобів віртуалізації для навчання ОС. Анкета містила як запитання щодо заздалегідь сформульованих факторів, так і можливість додати власні фактори. Оцінювання важливості факторів здійснювалося за шестибальною шкалою:

- «5» – дуже важливо;
- «4» – важливо;
- «3» – не обов'язково, але доречно;
- «2» – не дуже важливо;
- «1» – взагалі не важливо;
- «0» – важко відповісти.

Фактори, які оцінювалися, та розподіл оцінок респондентів наведено на рис. 4.13.

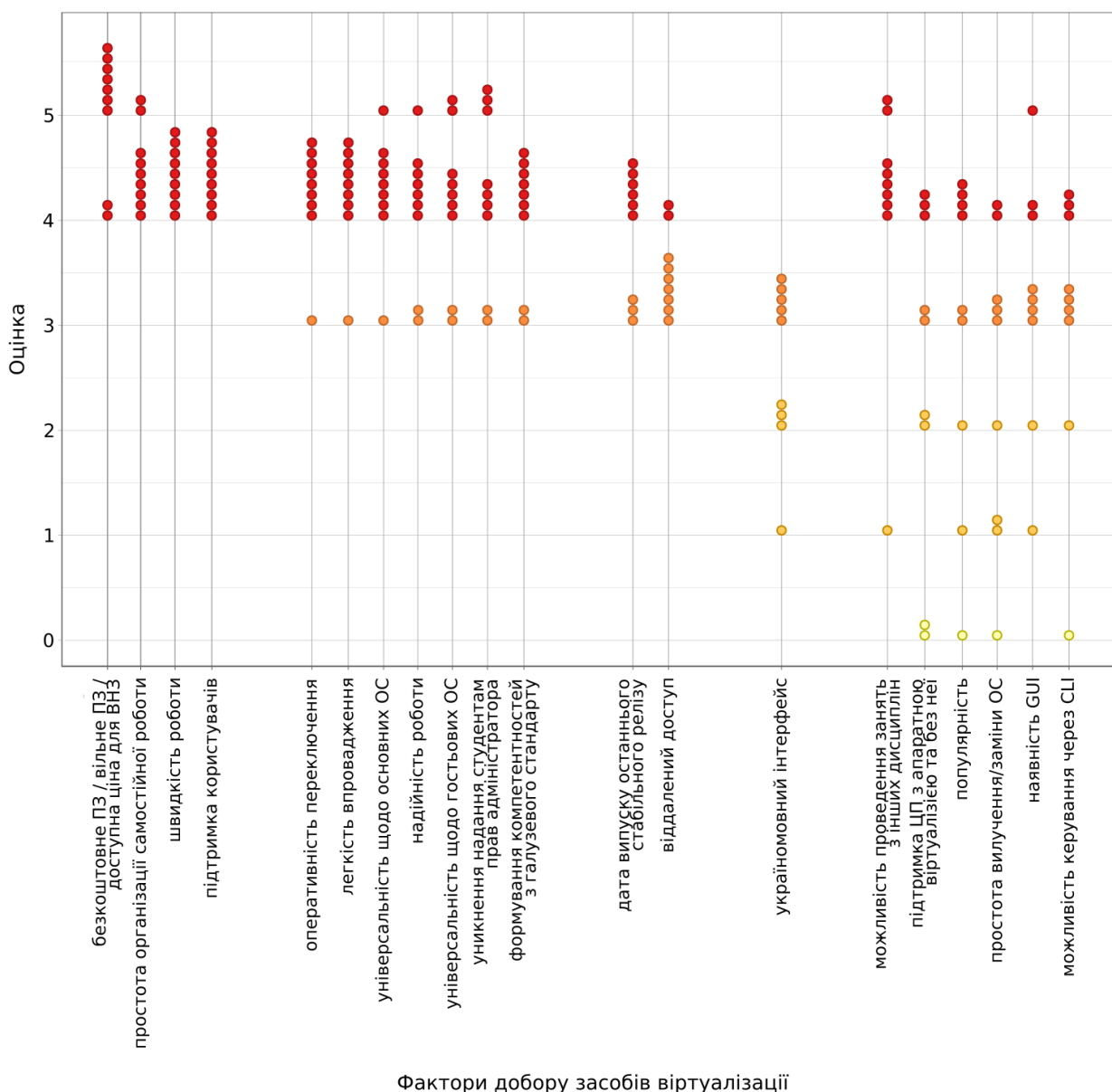


Рис. 4.13. Опитування викладачів курсу з ОС: розподіл відповідей респондентів за оцінюванням важливості факторів добору засобів віртуалізації

Залежно від одностайності респондентів під час оцінювання фактори можна умовно поділити на п'ять груп.

Група 1. Висока одностайність – висока важливість (на рис 4.13 – перша група зліва). Фактори з цієї групи отримали винятково оцінки «4» (важливо) і «5» (дуже важливо). До групи 1 увійшли такі фактори, як належність програмного засобу до безкоштовного, умовно безкоштовного, вільно поширюваного ПЗ або наявність спеціальних доступних пропозицій для навчальних закладів; простота організації студентської самостійної роботи з ОС, що вивчається; швидкість роботи; наявність різноманітних форм

безкоштовної підтримки користувачів програмного засобу (користувацька та технічна документація, форуми, списки розсилки тощо).

Група 2. Середня одностайність – висока і середня важливість (на рис. 4.13 – друга група зліва). Фактори з групи 2 також отримали більшість оцінок «4» (важливо) і «5» (дуже важливо), однак кожний з них одержав також одну-дві оцінки «3» (не обов'язково, але доречно). До групи ввійшли фактори: оперативність переключення між різними ОС; можливість без значних організаційних зусиль впровадити програмний засіб у навчальний процес; можливість встановлення програмного засобу віртуалізації поверх різних ОС (універсальність щодо основних ОС); надійність роботи; можливість віртуалізації різних ОС (універсальність щодо гостьових ОС); уникнення надання студенту прав адміністратора; можливість формування компетентностей, визначених галузевим стандартом.

Група 3. Середня одностайність – середня важливість (на рис. 4.13 – третя група зліва). Фактори даної групи набрали одержали оцінки «4» (важливо) та принаймні три оцінки «3» (не обов'язково, але доречно). До групи 3 потрапили фактори: дата випуску останнього стабільного релізу (актуальність програмного засобу); можливість віддаленого доступу до віртуалізованої ОС.

Група 4. Середня одностайність – середня і низька важливість (на рис. 4.13 – друга група справа). До цієї групи належить один фактор – наявність у програмного засобу україномовного інтерфейсу, який отримав оцінки «3» (не обов'язково, але доречно), «2» (не дуже важливо), «1» (взагалі не важливо).

Група 5. Низька одностайність (на рис. 4.13 – перша група справа). Фактори даної групи одержали найменш одностайні оцінки респондентів. Загалом оцінки факторів з групи 5 коливаються у діапазоні від «5» (дуже важливо) до «1» (взагалі не важливо), а також можуть набувати значення «0» (важко відповісти). В окремих випадках діапазон коливання може бути дещо меншим, але лишається значним. До групи 5 потрапили фактори: можливість проведення у комп'ютерному класі занять з інших дисциплін; підтримка

процесорів як із вбудованою апаратною віртуалізацією, так і без неї; популярність програмного засобу; простота вилучення/заміни ОС, що вивчається; наявність у програмному засобі графічного інтерфейсу; можливість керування програмним засобом із командного рядка.

Також в анкеті пропонувалося оцінити співвідношення використання графічного інтерфейсу (GUI) й інтерфейсу командного рядка (CLI) під час лабораторних робіт загалом, а також співвідношення їх застосування під час лабораторних робіт з окремих тем, оскільки воно може відрізнятися від загального співвідношення. Одержані відповіді відображено на рис. 4.14. Кілька крапок навпроти одного спостереження означають, що той самий респондент визначив використання різних інтерфейсів чи різне співвідношення їх поєднання для різних лабораторних робіт.

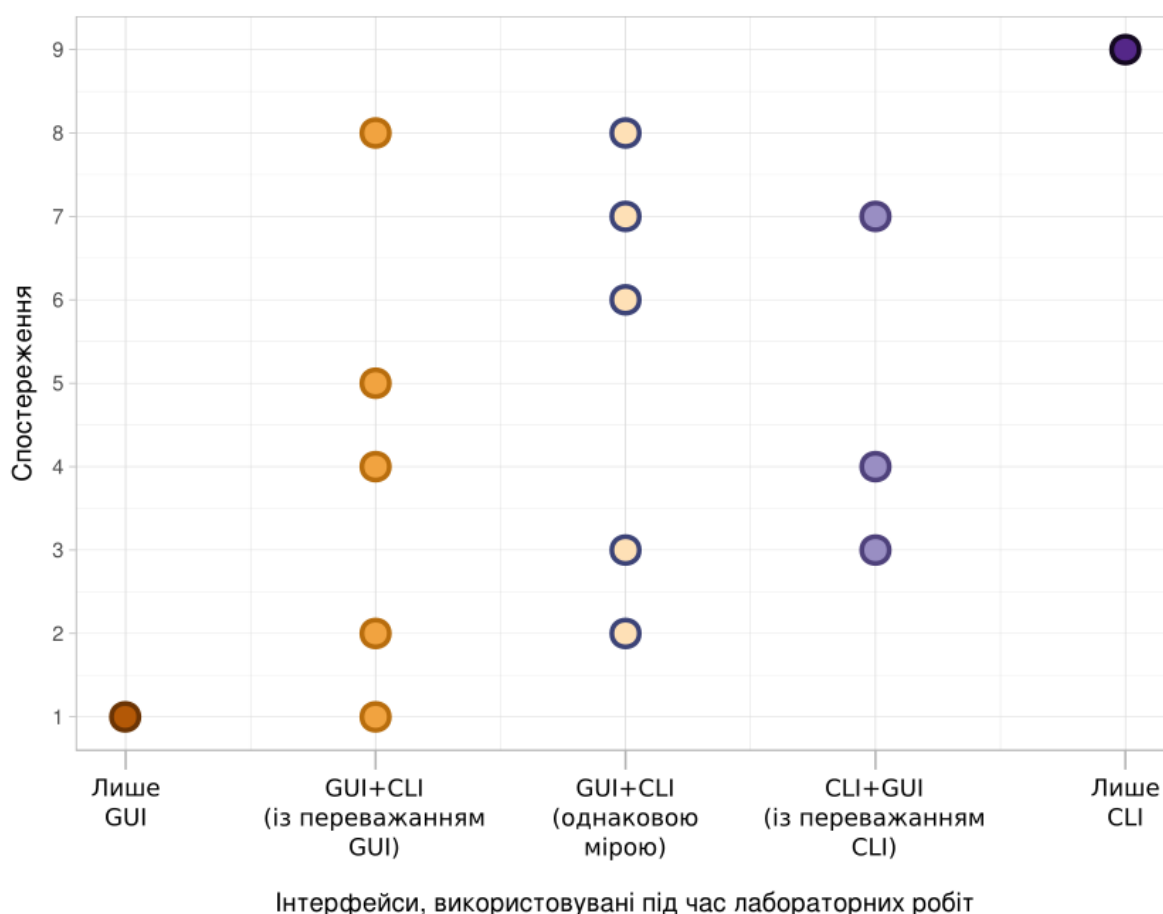


Рис. 4.14. Опитування викладачів курсу з ОС: розподіл відповідей респондентів за співвідношенням використання GUI/CLI під час лабораторних робіт

Загалом відповіді викладачів, які працювали з майбутніми бакалаврами інформатики, розміщені на діаграмі ближче до лівого краю (переважання графічного інтерфейсу), а відповіді викладачів, які працювали з майбутніми інженерами, – ближче до правого краю (переважання інтерфейсу командного рядка). Водночас єдиний випадок у вибірці, коли вказувалася робота лише з інтерфейсом командного рядка, стосувався саме бакалаврів інформатики. Та здебільшого спостерігається поєднання обох інтерфейсів у тій чи іншій пропорції.

Таким чином, у ході аналізу результатів анкетування викладачів курсу з ОС зроблено висновки про наявність низки відмінностей у навчанні ОС у ЗВО України. Особливо суттєво відрізнялися:

- ОС, що вивчалися;
- категорії використовуваного віртуалізаційного ПЗ та конкретні засоби віртуалізації;
- досвід застосування викладачами засобів віртуалізації;
- технічні умови, зокрема доступні викладачам альтернативи засобам віртуалізації та наявність процесорів із підтримкою апаратної віртуалізації;
- співвідношення застосування під час лабораторних робіт графічного інтерфейсу користувача та інтерфейсу командного рядка;
- оцінювання викладачами низки факторів добору засобів віртуалізації.

Результати анкетування викладачів, разом із результатами порівняльного аналізу засобів віртуалізації, проведеного у п. 2.3, привели до висновку про необхідність індивідуального добору засобів віртуалізації для курсу з ОС. Водночас, відзначені на основі бесід з викладачами та студентами й власного досвіду викладання курсу з ОС проблеми частих апаратних і програмних збоїв та неоднаковості умов, у яких здійснюється самостійна робота студентів, підштовхують до залучення альтернативних засобів віртуалізації поряд з основними, причому такий прийом знаходить підтвердження у закордонному досвіді викладання ОС.

Індивідуальний добір засобів віртуалізації для курсу з ОС та застосування альтернативних засобів віртуалізації було покладено в основу *варіативного підходу до застосування засобів віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики*, спираючись на який було розроблено методику застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики.

Наприкінці констатувального етапу педагогічного експерименту було уточнено гіпотезу дослідження шляхом зазначення в її формулюванні компетентностей, на покращення процесу формування яких спрямована розроблювана методика (інформатичні компетентності бакалаврів інформатики щодо ОС), а також здійснено підбір експериментальної бази для формувального етапу експерименту.

У ролі експериментальної бази для формувального етапу обрано Житомирський державний університет імені Івана Франка, Дрогобицький державний педагогічний університет імені Івана Франка, Сумський державний педагогічний університет імені А. С. Макаренка (майбутні бакалаври інформатики, на момент проведення експерименту – студенти напряму підготовки 6.040302 «Інформатика»). Обсяг контрольної групи (КГ) склав 53 особи, експериментальної групи (ЕГ) – 50 осіб, разом – 103 студенти (див. табл. 4.2).

Таблиця 4.2

Склад контрольної та експериментальної груп

<i>Експериментальна група</i>	<i>Контрольна група</i>
Житомирський державний університет імені Івана Франка, 2015-16 н. р., II курс, 24 чол., дисципліна «Операційні системи та системне програмування»	Житомирський державний університет імені Івана Франка, 2014-15 н. р., II курс, 25 чол., дисципліна «Операційні системи та системне програмування»
Сумський державний педагогічний університет імені А. С. Макаренка, 2015-16 н. р., III курс (18 чол.) і IV курс (8 чол.), дисципліна «Інформаційні системи»	Дрогобицький державний педагогічний університет імені Івана Франка: 2014-15 н. р., III курс (9 чол.), а також 2015-16 н. р., III курс (10 чол.) і IV курс (9 чол.), дисципліна «Операційні системи та системне програмування»

Однорідність вибірок перевірено шляхом аналізу наступних ознак:

- середнього балу, одержаного студентами на попередніх сесіях;

- частки студентів, які мали досвід користування Unix-подібними ОС до початку проходження курсу;

- частки студентів, які мали досвід користування засобами віртуалізації (віртуальними машинами, віртуальними контейнерами) до початку проходження курсу.

Для аналізу середніх балів, одержаних студентами під час попередніх сесій, застосовувався критерій Колмогорова-Смирнова. Вибір критерію тут і далі обумовлено обсягом вибірок, кількістю розрядів у розподілах, належністю критерію до непараметричних (завдяки чому розподіли не обов'язково мають бути наближеними до нормального), розрізнення розрядів за порядковою шкалою, а також потребою знаходження точки максимального накопичення розбіжностей.

Частоти розподілів за рівнями навчальних досягнень подано у табл. 4.3 та на рис. 4.15.

Таблиця 4.3

**Розподіли за рівнями навчальних досягнень
на основі середнього балу за попередні сесії**

Рівні навчальних досягнень	Контрольна група		Експериментальна група	
	Кількість, чол.	%	Кількість, чол.	%
Середній рівень («3»)	30	56,60%	25	50,00%
Достатній рівень («4»)	19	35,85%	20	40,00%
Високий рівень («5»)	4	7,55%	5	10,00%

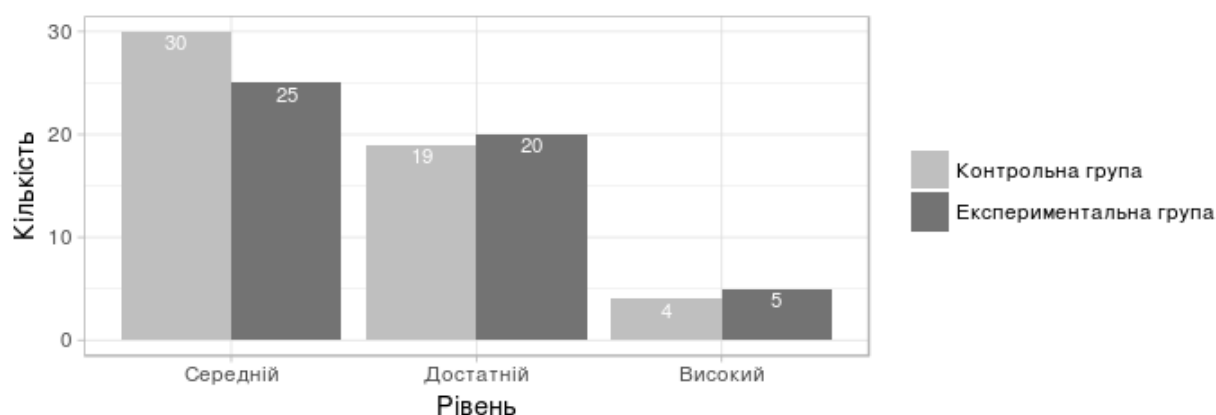


Рис. 4.15. Розподіли за рівнями навчальних досягнень на основі середнього балу за попередні сесії

Проаналізуємо дані табл. 4.3 за критерієм типу Колмогорова-Смирнова.

Критерій типу Колмогорова-Смирнова (λ -критерій) використовується для співставлення двох розподілів (у даному випадку – двох емпіричних розподілів) з метою відшукування точки, в якій сума накопичених розходжень між цими розподілами є найбільшою, та оцінювання достовірності цього розходження. Для правомірності застосування критерію необхідно, аби кожна з вибірок налічувала принаймні 50 спостережень, а також щоб розряди, за якими здійснюється вимірювання досліджуваної ознаки, були впорядкованими [152, с. 142]. Наявні вибірки ($n_e=50$, $n_k=53$, чотири впорядковані розряди) відповідають вказаним вимогам.

Сформулюємо статистичні гіпотези:

H_0 – розподіли за рівнями навчальних досягнень на основі середнього балу за попередні сесії у КГ та ЕГ однакові;

H_1 – розподіли за рівнями навчальних досягнень на основі середнього балу за попередні сесії у КГ та ЕГ відрізняються.

У табл. 4.4 наведено розрахунки емпіричних, відносних емпіричних та накопичених відносних емпіричних частот для обох розподілів, а також абсолютних різниць між накопиченими відносними частотами, де індекс e відповідає ЕГ, а індекс k – КГ.

Таблиця 4.4

**Розрахунок λ -критерію у процесі зіставлення емпіричних розподілів
за рівнями навчальних досягнень у контрольній та
експериментальній групах**

Рівні навчальних досягнень	Емпіричні частоти		Емпіричні відносні частоти		Накопичені емпіричні відносні частоти		Абсолютна різниця
	F_e	F_k	f_e	f_k	Σf_e	Σf_k	
Середній	25	30	0,500	0,566	0,500	0,566	0,066
Достатній	20	19	0,400	0,358	0,900	0,925	0,025
Високий	5	4	0,100	0,075	1,000	1,000	0,000
Всього	50	53	1,000	1,000			

Максимальна різниця між двома відповідними накопиченими відносними частотами (останній стовпчик табл. 4.4) складає $d_{\max}=0,066$. Обчислимо значення λ за наступною формулою:

$$\lambda = d_{max} \cdot \sqrt{\frac{n_e \cdot n_k}{n_e + n_k}} \quad (4.1)$$

Згідно з формулою 4.1,

$$\lambda = 0,066 \cdot \sqrt{\frac{50 \cdot 53}{50 + 53}} \approx 0,066 \cdot 5,072 \approx 0,335$$

Отримане емпіричне значення λ менше за критичне значення $\lambda_{0,05} = 1,224$ для порівняння двох емпіричних розподілів [240, с. 51] (рис. 4.16).

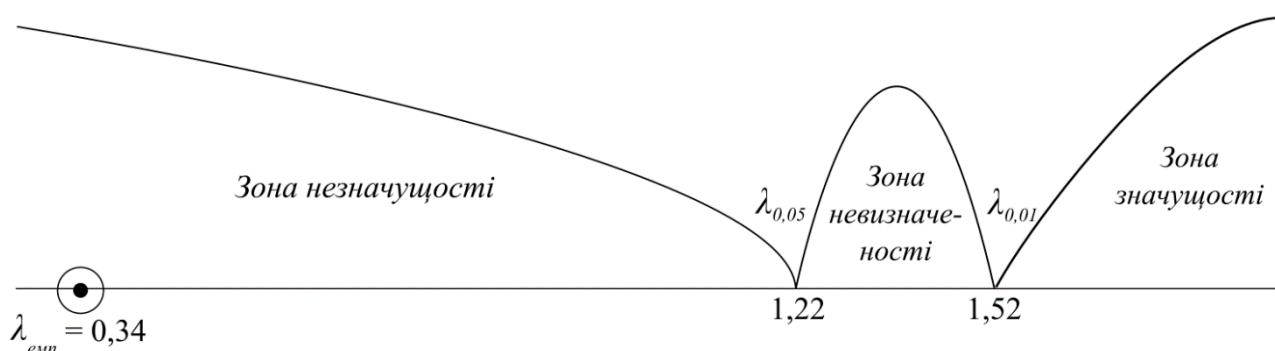


Рис. 4.16. Вісь значущості для емпіричного значення критерію типу Колмогорова-Смирнова для розподілів за рівнями навчальних досягнень на основі середнього балу за попередні сесії

Робимо висновок про статистичну незначущість розбіжностей між розподілами для рівня значущості $\leq 0,05$.

Маємо право відхилити гіпотезу H_1 і прийняти гіпотезу H_0 : розподіли за рівнями навчальних досягнень на основі середнього балу за попередні сесії у КГ та ЕГ однакові.

Відомості про частку студентів, які ще до початку вивчення дисципліни мали досвід користування Unix-подібними ОС, та студентів, які на той самий момент уже працювали із засобами віртуалізації, було зібрано шляхом усного опитування на перших заняттях з дисципліни.

Питання було сформульовано наступним чином:

«Чи хтось із вас уже працював з Linux чи іншою Unix-подібною операційною системою раніше?» (будь-який досвід роботи, навіть мінімальний, вважався відповіддю «так»)

«Чи вже хтось із вас використовував віртуальні машини? віртуальні контейнери?» (як і для попереднього питання, відповіддю «так» вважався будь-

який досвід роботи)

Важливо зазначити, що опитування здійснювалося після ознайомлення студентів з поняттями, які фігурують у питаннях (Unix-подібні ОС, ядро і дистрибутиви Linux, віртуалізація). У ході спілкування запитання викладача та відповіді студентів за потреби уточнювалися. Наприклад, робота з мобільною ОС Android на рівні користувача за відповідь «так» не вважалася. Розподіл відповідей студентів із контрольної та експериментальної груп на запитання про досвід роботи з Unix-подібними ОС подано у табл. 4.5 та на рис. 4.17.

Таблиця 4.5

**Розподіл відповідей студентів на запитання
про досвід роботи з Unix-подібними ОС**

Групи	Емпіричні частоти вибору		Всього
	Експериментальна група	Контрольна група	
Досвід наявний	6	5	11
Досвід відсутній	44	48	92
Всього	50	53	103

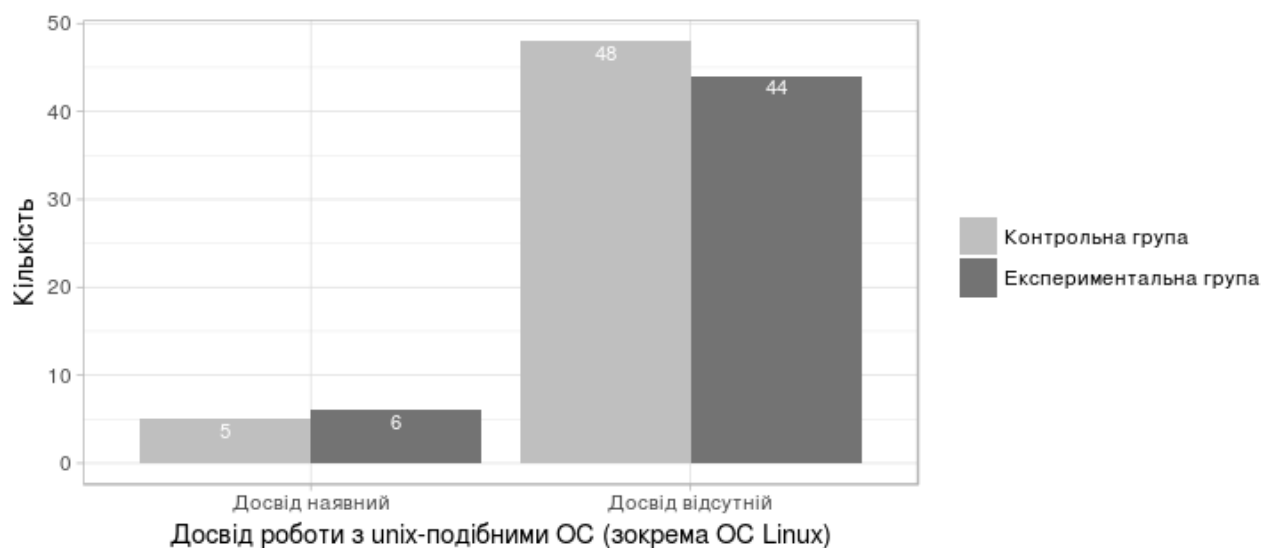


Рис. 4.17. Розподіли за досвідом роботи з Unix-подібними ОС до початку вивчення дисципліни

Для співставлення результатів, одержаних для КГ та ЕГ, застосуємо кутове перетворення Фішера. Вибір критерію тут і далі обумовлено кількістю рівнів у розподілах, належністю критерію до непараметричних (завдяки чому розподіли не обов'язково мають бути наближеними до нормального).

Кутове перетворення Фішера (ф-критерій, кутове перетворення Фішера) призначений для співставлення двох вибірок відповідно до частоти спостереження деякого ефекту, що становить інтерес для дослідника, і полягає в оцінюванні достовірності розбіжностей між тими процентними частками двох вибірок, в яких зареєстровано цей ефект. Для використання критерію Фішера необхідно, щоб жодна з процентних часток, про які йшлося вище, не була рівною нулю. У разі, коли кожна з аналізованих вибірок налічує принаймні 5 спостережень, співвідношення між конкретними значеннями n_e та n_k дозволяються будь-які [152, с. 160].

Для застосування кутового перетворення Фішера спостереження ділять на дві підгрупи: спостереження, у яких спостерігалось деяке досліджуване явище (проявлявся деякий ефект), та спостереження, у яких досліджуване явище не спостерігалось (ефект не проявлявся). У даному випадку у ролі досліджуваного явища виступає наявність у студентів попереднього досвіду роботи з Unix-подібними ОС, відповідно спостереження, у котрих студенти мали такий досвід, віднесемо до підгрупи «явище спостерігається», а спостереження, у яких такий досвід у студентів був відсутній, – до підгрупи «явище не спостерігається».

Складемо чотириклітинну таблицю (табл. 4.6) для розрахунку кутового перетворення Фішера для виявлення відмінностей між підгрупами «явище спостерігається» та «явище не спостерігається» у КГ та ЕГ.

Таблиця 4.6

**Чотириклітинна таблиця для розрахунку ф-критерію Фішера
за результатами опитування студентів
щодо їхнього попереднього досвіду роботи з Unix-подібними ОС**

Група	«Явище спостерігається»		«Явище не спостерігається»		Всього
	кількість	%	кількість	%	
Експериментальна	32	64,0%	18	36,0%	50
Контрольна	19	35,8%	34	64,2%	53

Сформулюємо статистичні гіпотези:

H_0 – розподіли за наявністю попереднього досвіду роботи з Unix-подібними ОС у КГ та ЕГ однакові;

H_1 – розподіли за наявності попереднього досвіду роботи з Unix-подібними ОС у КГ та ЕГ відрізняються.

За таблицями з [152, с. 330], визначаємо величини кута φ , що відповідають процентним часткам спостережень з підгрупи «явище спостерігається» в КГ та ЕГ. Для ЕГ маємо $\varphi_1(12,0\%)=0,707$, для КГ – $\varphi_2(9,4\%)=0,632$.

Розраховуємо емпіричне значення критерію φ^* за формулою:

$$\varphi_{\text{емп}}^* = (\varphi_1 - \varphi_2) \cdot \sqrt{\frac{n_e \cdot n_k}{n_e + n_k}}, \quad (4.2)$$

де кут φ_1 відповідає більшій, а кут φ_2 – меншій процентній частці.

Згідно з формулою 4.2,

$$\varphi_{\text{емп}}^* = (0,707 - 0,632) \cdot \sqrt{\frac{50 \cdot 53}{50 + 53}} \approx 0,075 \cdot 5,072 \approx 0,38$$

Критичні значення для кутового перетворення Фішера дорівнюють

$$\varphi_{\text{кр}}^* = \begin{cases} 1,64 & (\rho \leq 0,05) \\ 2,31 & (\rho \leq 0,01) \end{cases}$$

Отримане значення $\varphi_{\text{емп}}^*$ менше за критичне для рівня значущості 0,05. Побудуємо вісь значущості (рис. 4.18).

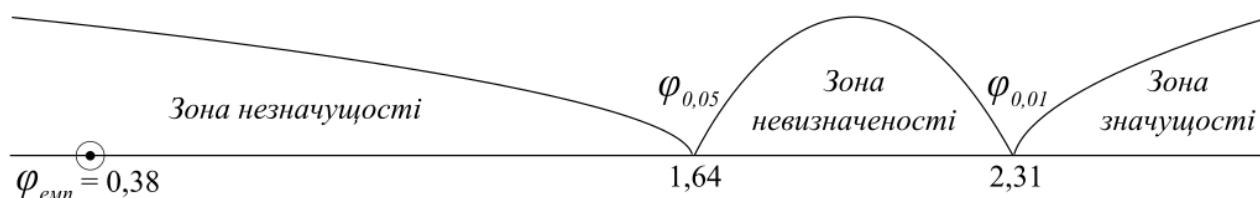


Рис. 4.18. Вісь значущості для емпіричного значення кутового перетворення Фішера для розподілів за досвідом роботи з Unix-подібними ОС

Обчислене значення $\varphi_{\text{емп}}^*$ міститься у зоні незначущості. Маємо право відхилити гіпотезу H_1 і прийняти гіпотезу H_0 : розподіли за наявності попереднього досвіду роботи з Unix-подібними ОС у КГ та ЕГ однакові.

Також здійснено опрацювання одержаних даних засобами мови статистичного виведення R. Для даних таблиці 4.5 виконано двосторонній тест Фішера (функція `fisher.test`). Отримано p-значення (p-value) $p\text{-value} = 0,7561$, що значно перевищує 0,05, при довірчому інтервалі для рівня довіри 95% від

0,307602 до 5,817392, який містить одиницю, що підтверджує зроблені раніше висновки про відхилення гіпотези H_1 і прийняття гіпотези H_0 : розподіли за рівнями навчальних досягнень у КГ та ЕГ співпадають. Повні програмні коди мовою R, а також отримані результати подано у додатку С.

Розподіл відповідей піддослідних студентів на запитання про досвід роботи із засобами віртуалізації наведено у табл. 4.7. та на рис. 4.19.

Таблиця 4.7

**Розподіл відповідей студентів на запитання
про досвід роботи із засобами віртуалізації**

Групи	Емпіричні частоти вибору		Всього
	Експериментальна група	Контрольна група	
Досвід наявний	4	6	10
Досвід відсутній	46	47	93
Всього	50	53	103

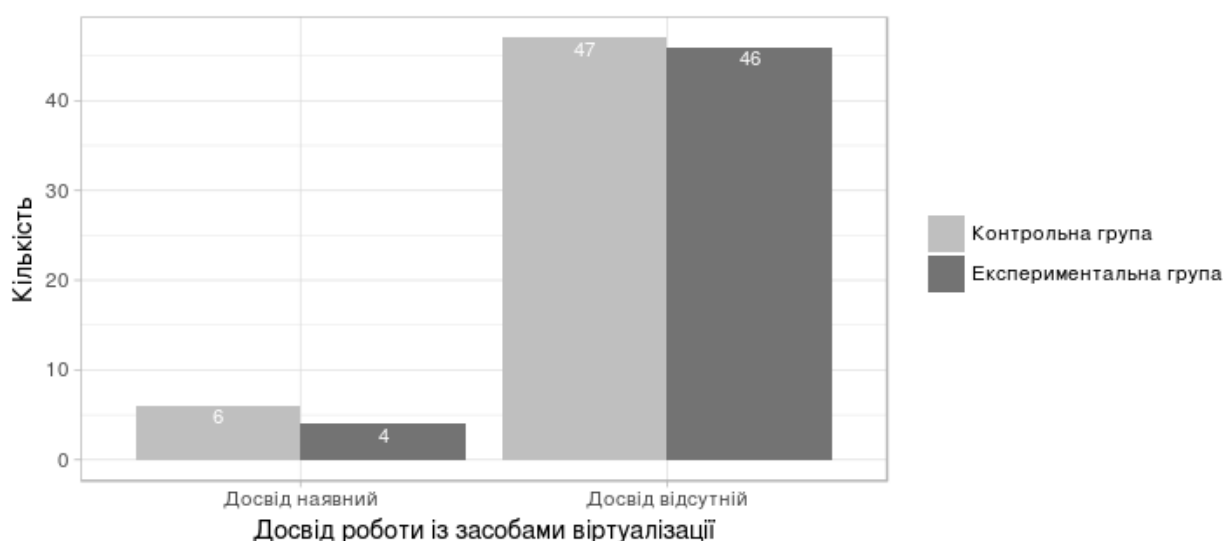


Рис. 4.19. Розподіли за досвідом роботи із засобами віртуалізації до початку вивчення дисципліни

Щоб з'ясувати, чи існують статистично значущі розбіжності між розподілами відповідей у КГ та ЕГ, використаємо кутове перетворення Фішера.

Досліджуваним явищем у цьому випадку є наявність у студентів попереднього досвіду роботи із засобами віртуалізації. Спостереження, у яких студенти мали такий досвід, належать до підгрупи «явище спостерігається», а спостереження, у яких такий досвід у студентів відсутній, віднесемо до

підгрупи «явище не спостерігається».

Складемо чотириклітинну таблицю (табл. 4.8) для обчислення кутового перетворення Фішера для виявлення відмінностей між підгрупами «явище спостерігається» та «явище не спостерігається» у КГ та ЕГ.

Таблиця 4.8

**Чотириклітинна таблиця для розрахунку ф-критерію Фішера
за результатами опитування студентів
щодо їхнього попереднього досвіду роботи із засобами віртуалізації**

Група	«Явище спостерігається»		«Явище не спостерігається»		Всього
	кількість	%	кількість	%	
Експериментальна	4	8,0%	46	92,0%	50
Контрольна	6	11,3%	47	88,7%	53

Сформулюємо статистичні гіпотези:

H_0 – розподіли за наявністю попереднього досвіду роботи із засобами віртуалізації у КГ та ЕГ однакові;

H_1 – розподіли за наявністю попереднього досвіду роботи із засобами віртуалізації у КГ та ЕГ відрізняються.

За таблицями, наведеними у [152, с. 330], визначаємо величини кута ϕ , що відповідають процентним часткам спостережень з підгрупи «явище спостерігається» в КГ та ЕГ. Для КГ маємо $\phi_1(11,3\%)=0,686$, для ЕГ – $\phi_2(8,0\%)=0,574$ (кут ϕ_1 відповідає більшому куту, тому порядок тут відмінний від порядку, використаного для даних табл. 3.6).

Розраховуємо емпіричне значення критерію ϕ^* за формулою (4.2):

$$\phi_{\text{емп}}^* = (0,686 - 0,574) \cdot \sqrt{\frac{50 \cdot 53}{50 + 53}} \approx 0,139 \cdot 5,072 \approx 0,57$$

Критичні значення для кутового перетворення Фішера дорівнюють

$$\phi_{\text{кр}}^* = \begin{cases} 1,64 & (\rho \leq 0,05) \\ 2,31 & (\rho \leq 0,01) \end{cases}$$

Отримане значення $\phi_{\text{емп}}^*$ менше за критичне для рівня значущості 0,05. Побудуємо вісь значущості (рис. 4.20).

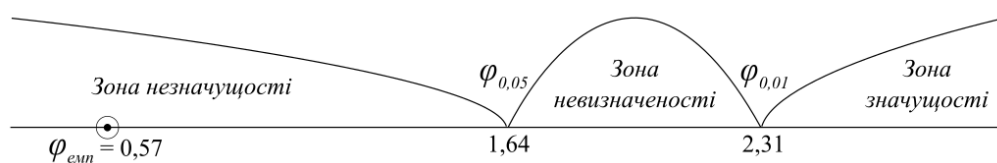


Рис. 4.20. Вісь значущості для емпіричного значення кутового перетворення Фішера для розподілів за досвідом роботи із засобами віртуалізації

Обчислене значення $\varphi_{\text{емп}}^*$ міститься у зоні незначущості. Маємо право відхилити гіпотезу H_1 і прийняти гіпотезу H_0 : розподіли за наявностію попереднього досвіду роботи із засобами віртуалізації у КГ та ЕГ однакові.

Також здійснено опрацювання одержаних даних засобами мови статистичного виведення R. Для даних таблиці 4.7 виконано двосторонній тест Фішера (функція `fisher.test`). Отримано р-значення (p-value) $p\text{-value} = 0,7423$, що значно перевищує 0,05, при довірчому інтервалі для рівня довіри 95% від 0,132864 до 3,100517, який містить одиницю, що підтверджує зроблені раніше висновки про відхилення гіпотези H_1 і прийняття гіпотези H_0 : розподіли за рівнями навчальних досягнень у КГ та ЕГ співпадають. Повні програмні коди мовою R, а також отримані результати подано у додатку С.

Таким чином, однорідність вибірок перевірено шляхом аналізу середнього балу, одержаного студентами на попередніх сесіях; частки студентів, які мали досвід користування Unix-подібними ОС до початку проходження курсу; частки студентів, які мали досвід користування засобами віртуалізації (віртуальними машинами, віртуальними контейнерами) до початку проходження курсу. В усіх трьох випадках не виявлено статистично значущих відмінностей між розподілами досліджуваних ознак у контрольній та експериментальній групах. Отже, однорідність вибірок підтверджено на рівні значущості 0,05.

4.3. Проведення та аналіз результатів формувального етапу експерименту

Метою формувального етапу педагогічного експерименту було підтвердити або спростувати гіпотезу дослідження про підвищення результативності формування інформатичних компетентностей бакалаврів

інформатики щодо ОС у разі, якщо навчання Unix-подібних операційних систем здійснюватиметься на основі розробленої методики застосування технологій віртуалізації.

Формувальний етап педагогічного експерименту проводився у період з 2015 по 2018 рік на базі Житомирського державного університету імені Івана Франка, Дрогобицького державного педагогічного університету імені Івана Франка, Сумського державного педагогічного університету імені А. С. Макаренка з майбутніми бакалаврами інформатики (напряму підготовки 6.040302 «Інформатика»). Обсяг КГ – 53 особи, обсяг ЕГ – 50 осіб, разом 103 осіб. Склад КГ і ЕГ та перевірку однорідності вибірок описано у п. 4.1.

У ході проведення формувального етапу експерименту навчання студентів ЕГ здійснювалося на основі пропонованої методики застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики, тимчасом як для навчання студентів КГ зазначена методика не використовувалася.

Було одержано та проаналізовано наступні бали студентів:

- середній бал, одержаний за захист лабораторних робіт (сума балів за всі захищені роботи, поділені на загальну кількість передбачених курсом лабораторних робіт);
- бали, набрані під час проходження двох контрольних тестових опитувань з теорії ОС (додатки Т і У);
- бал, набраний під час проходження тестового опитування «Unix-подібні операційні системи» (додаток П).

Необхідність проведення окремого опитування, присвяченого саме Unix-подібним ОС, обґрунтовано у п. 4.2.

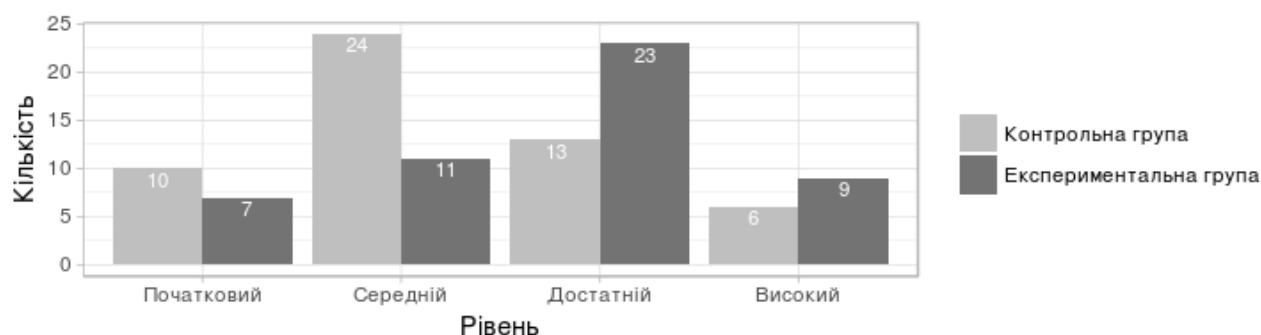
Оцінювання відбувалося за 100-бальною шкалою (від 0 до 100 балів) із переведенням у бал за національною шкалою (від 2 до 5 балів) й відповідно до критеріїв, детально описаних у п. 3.2.

Проаналізуємо результати **тестового опитування «Unix-подібні операційні системи»**. Частоти розподілів за рівнями навчальних досягнень подано у таблиці 4.9 та на рис. 4.21.

Таблиця 4.9

**Розподіли за рівнями навчальних досягнень за результатами
тестового опитування «Unix-подібні операційні системи»**

Рівні навчальних досягнень	Контрольна група		Експериментальна група	
	Кількість, чол.	%	Кількість, чол.	%
Початковий рівень	10	18,87%	7	14,00%
Середній рівень	24	45,28%	11	22,00%
Достатній рівень	13	24,53%	23	46,00%
Високий рівень	6	11,32%	9	18,00%



*Рис. 4.21. Розподіли за рівнями навчальних досягнень
на основі результатів тестового опитування «Unix-подібні ОС»*

Проаналізуємо дані з табл. 4.9 за допомогою критерію типу Колмогорова-Смирнова. У табл. 4.10 наведено розрахунки емпіричних, відносних емпіричних та накопичених відносних емпіричних частот для обох розподілів, а також абсолютних різниць між накопиченими відносними частотами, де індекс e відповідає ЕГ, а індекс k – КГ.

Таблиця 4.10

**Розрахунок λ -критерію у процесі зіставлення емпіричних розподілів
за рівнями навчальних досягнень у контрольній та
експериментальній групах**

Рівні навчальних досягнень	Емпіричні частоти		Емпіричні відносні частоти		Накопичені емпіричні відносні частоти		Абсолютна різниця
	F_e	F_k	f_e	f_k	Σf_e	Σf_k	
Початковий	7	10	0,140	0,189	0,140	0,189	0,049
Середній	11	24	0,220	0,453	0,360	0,642	0,282
Достатній	23	13	0,460	0,245	0,820	0,887	0,067
Високий	9	6	0,180	0,113	1,000	1,000	0,000
Всього	50	53	1,000	1,000			

Максимальна різниця між двома відповідними накопиченими відносними частотами (останній стовпчик табл. 4.10) складає $d_{\max}=0,282$. Обчислимо значення λ за формулою (4.1):

$$\lambda = 0,282 \cdot \sqrt{\frac{50 \cdot 53}{50 + 53}} \approx 0,282 \cdot 5,072 \approx 1,43$$

Згідно з [240, с. 51], критичні значення для порівняння двох емпіричних розподілів для рівнів значущості 0,05 та 0,01 становлять 1,224 та 1,517 відповідно. Отримане емпіричне значення λ більше за критичне $\lambda_{0,05} = 1,224$, але менше за критичне $\lambda_{0,01} = 1,517$ (рис. 4.22).

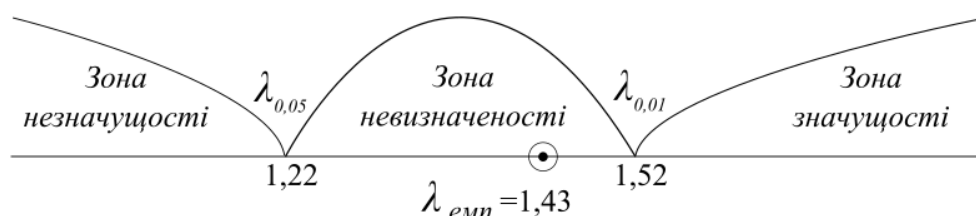


Рис. 4.22. Вісь значущості для емпіричного значення критерію типу Колмогорова-Смирнова для розподілів за результатами тестового опитування «Unix-подібні ОС»

Для перевірки достовірності результату скористаємось ф-критерієм Фішера.

У ролі досліджуваного явища у цьому випадку виступає наявність чи відсутність позитивного ефекту від організації процесу навчання студентів з КГ та ЕГ, відповідно, без та із застосуванням запропонованої методики. Розділимо аналізовані вибірки на дві підгрупи: «є ефект» та «немає ефекту». На відміну від попередніх випадків застосування ф-критерію Фішера, у табл. 4.10 маємо не два, а чотири рівні ознаки, і загалом можливі різні способи розбиття їх на дві групи. Для одержання якнайточніших результатів поділ здійснюватимемо у точці максимального розходження між вибірками – точці $d_{\max}$, значення якої було обчислене під час розрахунку критерію Колмогорова-Смирнова, становить $d_{\max} = 0,282$ і відповідає середньому рівню навчальних досягнень. Отже, спостереження, у яких зареєстровано достатній та високий рівні навчальних досягнень, віднесемо до підгрупи «є ефект», а спостереження з початковим та середнім рівнем – до підгрупи «немає ефекту» (табл. 4.11).

Таблиця 4.11

Розподіл на підгрупи «є ефект» та «немає ефекту»

Групи	Емпіричні частоти вибору		Всього
	Експериментальна група	Контрольна група	
I група: достатній і високий рівні («є ефект»)	32	19	51
II група: початковий і середній рівні («немає ефекту»)	18	34	52
Всього	50	53	103

Складемо чотириклітинну таблицю (табл. 4.12) для розрахунку критерію Фішера для виявлення відмінностей між підгрупами «є ефект» та «немає ефекту» у КГ та ЕГ.

Таблиця 4.12

**Чотириклітинна таблиця для розрахунку ф-критерію Фішера
за результатами тестового опитування
«Unix-подібні операційні системи»**

Група	«Є ефект»		«Немає ефекту»		Всього
	кількість	%	кількість	%	
Експериментальна	32	64,0%	18	36,0%	50
Контрольна	19	35,8%	34	64,2%	53

Сформулюємо статистичні гіпотези:

H_0 – розподіли за рівнями навчальних досягнень у КГ та ЕГ однакові;

H_1 – розподіли за рівнями навчальних досягнень у КГ та ЕГ відрізняються.

За таблицями, наведеними у [152, с. 330], визначаємо величини кута φ , що відповідають процентним часткам спостережень з підгрупи «є ефект» в КГ та ЕГ. Для ЕГ маємо $\varphi_1(64,0\%)=1,855$, для КГ – $\varphi_2(35,8\%)=1,283$.

Розраховуємо емпіричне значення критерію φ^* за формулою 4.2:

$$\varphi_{\text{емп}}^* = (1,855 - 1,283) \cdot \sqrt{\frac{50 \cdot 53}{50 + 53}} \approx 0,572 \cdot 5,072 \approx 2,90$$

За таблицею [152, с. 332] обчислюємо, що одержане $\varphi_{\text{емп}}^*$ відповідає рівню значущості $\rho < 0,001$ (уточнене значення, розраховане засобами мови R,

становить 0,00577).

Критичні значення для φ -критерію Фішера дорівнюють

$$\varphi_{\text{кр}}^* = \begin{cases} 1,64 & (\rho \leq 0,05) \\ 2,31 & (\rho \leq 0,01) \end{cases}$$

Отримане значення $\varphi_{\text{емп}}^*$ більше за відповідне критичне: $2,90 > 2,31$, а отже, одержані результати достовірні для рівня значущості 0,01. Побудуємо вісь значущості (рис. 4.23).



Рис. 4.23. Вісь значущості для емпіричного значення φ -критерію Фішера для розподілів за результатами тестового опитування «Unix-подібні ОС»

Обчислене значення $\varphi_{\text{емп}}^*$ міститься у зоні значущості. Маємо право відхилити гіпотезу H_0 і прийняти гіпотезу H_1 : розподіли за рівнями навчальних досягнень у КГ та ЕГ відрізняються.

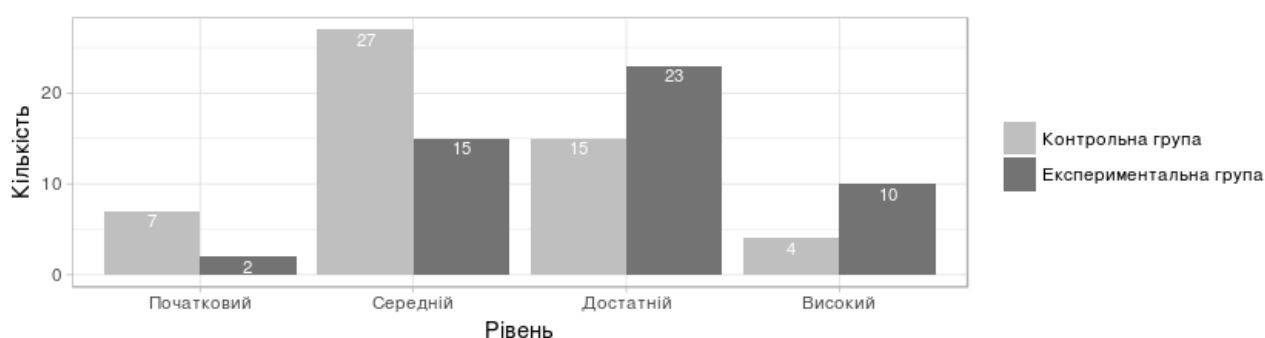
Паралельно було здійснено опрацювання одержаних даних засобами мови R. Для даних таблиці 4.3.12 виконано двосторонній тест Фішера (функція `fisher.test`). Одержано мале p-значення (p-value) $p\text{-value} = 0,005766$, високе співвідношенням шансів (odds ratio) $OR = 3,143687$ при довірчому інтервалі для рівня довіри 95% (95 percent confidence interval) від 1,322439 до 7,711034, який не містить одиниці, що підтверджує зроблені раніше висновки про відхилення гіпотези H_0 і прийняття гіпотези H_1 : розподіли за рівнями навчальних досягнень у КГ та ЕГ відрізняються. Повні програмні коди мовою R, а також отримані результати подано у додатку С.

Здійснимо аналіз **середнього балу за захист лабораторних робіт**. Частоти розподілів за рівнями навчальних досягнень студентів наведено у таблиці 4.13 та на рис. 4.24.

Таблиця 4.13

**Розподіли за рівнями навчальних досягнень
на основі середнього балу за захист лабораторних робіт**

Рівні навчальних досягнень	Контрольна група		Експериментальна група	
	Кількість, чол.	%	Кількість, чол.	%
Початковий рівень	7	13,21%	2	4,00%
Середній рівень	27	50,94%	15	30,00%
Достатній рівень	15	28,30%	23	46,00%
Високий рівень	4	7,55%	10	20,00%



*Рис. 4.24. Розподіли за рівнями навчальних досягнень
на основі середнього балу за захист лабораторних робіт*

Виконаємо аналіз одержаних даних за допомогою критерію типу Колмогорова-Смирнова (табл. 4.14).

Таблиця 4.14

**Розрахунок λ -критерію у процесі зіставлення емпіричних розподілів
за рівнями навчальних досягнень у контрольній та
експериментальній групах**

Рівні навчальних досягнень	Емпіричні частоти		Емпіричні відносні частоти		Накопичені емпіричні відносні частоти		Абсолютна різниця
	F_e	F_k	f_e	f_k	Σf_e	Σf_k	
Початковий	2	7	0,040	0,132	0,040	0,132	0,092
Середній	15	27	0,300	0,509	0,340	0,642	0,302
Достатній	23	15	0,460	0,283	0,800	0,925	0,125
Високий	10	4	0,200	0,075	1,000	1,000	0,000
Всього	50	53	1,000	1,000			

Максимальна різниця між двома відповідними накопиченими відносними частотами складає $d_{\max}=0,302$. Обчислимо значення λ за формулою 4.1:

$$\lambda = 0,302 \cdot \sqrt{\frac{50 \cdot 53}{50 + 53}} \approx 0,302 \cdot 5,072 \approx 1,53$$

Розраховане критичне значення λ більше за критичне $\lambda_{0,05} = 1,224$ і дещо перевищує критичне $\lambda_{0,01} = 1,517$ [240, с. 51] (рис. 4.25).



Рис. 4.25. Вісь значущості для емпіричного значення критерію типу Колмогорова-Смирнова для розподілів за середнім балом за захист лабораторних робіт

Для уточнення одержаного результату скористаємось ϕ -критерієм Фішера. Розподіл аналізованих вибірок на підгрупи «є ефект» і «немає ефекту» подано у табл. 4.15.

Таблиця 4.15

Розподіл на підгрупи «є ефект» та «немає ефекту»

Групи	Емпіричні частоти вибору		Всього
	Експериментальна група	Контрольна група	
I група: достатній і високий рівні («є ефект»)	33	19	52
II група: початковий і середній рівні («немає ефекту»)	17	34	51
Всього	50	53	103

Складемо чотириклітинну таблицю (табл. 4.16) для розрахунку кутового перетворення Фішера.

Таблиця 4.16

Чотириклітинна таблиця для розрахунку ϕ -критерію Фішера за середнім балом за захист лабораторних робіт

Група	«Є ефект»		«Немає ефекту»		Всього
	кількість	%	кількість	%	
Експериментальна	33	66,0%	17	34,0%	50
Контрольна	19	35,8%	34	64,2%	53

Сформулюємо статистичні гіпотези:

H_0 – розподіли за рівнями навчальних досягнень у КГ та ЕГ однакові;

H_1 – розподіли за рівнями навчальних досягнень у КГ та ЕГ відрізняються.

Величини кута φ , що відповідають процентним часткам спостережень з підгрупи «є ефект» в КГ та ЕГ, складають $\varphi_1(66,0\%)=1,897$ (ЕГ) і $\varphi_2(35,8\%)=1,283$ (КГ).

Обчислюємо емпіричне значення критерію φ^* за формулою 4.2:

$$\varphi_{\text{емп}}^* = (1,897 - 1,283) \cdot \sqrt{\frac{50 \cdot 53}{50 + 53}} \approx 0,614 \cdot 5,072 \approx 3,11$$

Одержане $\varphi_{\text{емп}}^*$ відповідає рівню значущості $\rho < 0,001$ (уточнене значення, розраховане засобами мови R, становить 0,00305).

Критичні значення для кутового перетворення Фішера дорівнюють

$$\varphi_{\text{кр}}^* = \begin{cases} 1,64 & (\rho \leq 0,05) \\ 2,31 & (\rho \leq 0,01) \end{cases}$$

Отримане значення $\varphi_{\text{емп}}^*$ більше за відповідне критичне: $3,11 > 2,31$, а отже, одержані результати достовірні для рівня значущості 0,01. Побудуємо вісь значущості (рис. 4.26).



Рис. 4.26. Вісь значущості для емпіричного значення φ -критерію Фішера для розподілів за середнім балом за захист лабораторних робіт

Одержане значення $\varphi_{\text{емп}}^*$ міститься у зоні значущості. Маємо право відхилити гіпотезу H_0 і прийняти гіпотезу H_1 : розподіли за рівнями навчальних досягнень у КГ та ЕГ відрізняються.

Було також здійснено опрацювання одержаних даних засобами мови R. Для даних таблиці 4.16 виконано двосторонній тест Фішера (функція `fisher.test`). Одержано мале р-значення $p\text{-value} = 0,003051$, високе співвідношенням шансів

OR = 3,429061 при довірчому інтервалі для рівня довіри 95% від 1,435398 до 8,478342, який не містить одиниці, що підтверджує зроблені раніше висновки про відхилення гіпотези H_0 і прийняття гіпотези H_1 : розподіли за рівнями навчальних досягнень у КГ та ЕГ відрізняються. Повні програмні коди мовою R, а також отримані результати подано у додатку С.

Виконаємо аналіз результатів **першого контрольного тестового опитування з теорії ОС**.

Частоти розподілів за рівнями навчальних досягнень подано у таблиці 4.17 та на рис. 4.27.

Таблиця 4.17

**Розподіли за рівнями навчальних досягнень
на основі результатів першого контрольного тестового опитування
з теорії ОС**

Рівні навчальних досягнень	Контрольна група		Експериментальна група	
	Кількість, чол.	%	Кількість, чол.	%
Початковий рівень	8	15,09%	4	8,00%
Середній рівень	28	52,83%	16	32,00%
Достатній рівень	16	30,19%	23	46,00%
Високий рівень	1	1,89%	7	14,00%

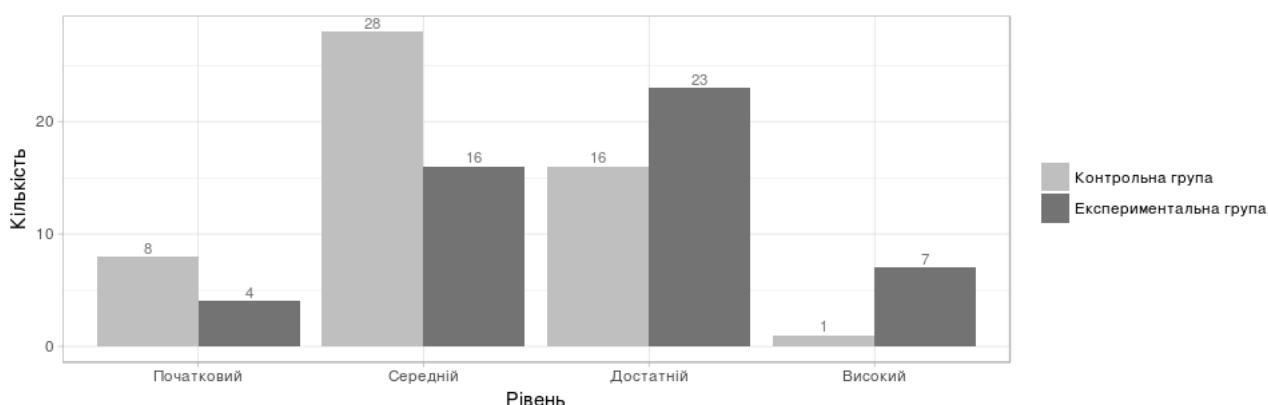


Рис. 4.27. Розподіли за рівнями навчальних досягнень на основі результатів першого контрольного тестового опитування з теорії ОС

Проаналізуємо дані за критерієм типу Колмогорова-Смирнова (табл. 4.18).

Таблиця 4.18

Розрахунок λ -критерію у процесі зіставлення емпіричних розподілів за рівнями навчальних досягнень у контрольній та експериментальній групах

Рівні навчальних досягнень	Емпіричні частоти		Емпіричні відносні частоти		Накопичені емпіричні відносні частоти		Абсолютна різниця
	F_e	F_k	f_e	f_k	Σf_e	Σf_k	
Початковий	4	8	0,080	0,151	0,080	0,151	0,071
Середній	16	28	0,320	0,528	0,400	0,679	0,279
Достатній	23	16	0,460	0,302	0,860	0,981	0,121
Високий	7	1	0,140	0,019	1,000	1,000	0,000
Всього	50	53	1,000	1,000			

Максимальна різниця між відповідними накопиченими відносними частотами складає $d_{\max}=0,279$. Обчислимо значення λ за формулою 4.1:

$$\lambda = 0,279 \cdot \sqrt{\frac{50 \cdot 53}{50 + 53}} \approx 0,279 \cdot 5,072 \approx 1,42$$

Розраховане критичне значення λ більше за критичне $\lambda_{0,05} = 1,224$, але менше за критичне $\lambda_{0,01} = 1,517$ [240, с. 51] (рис. 4.28).

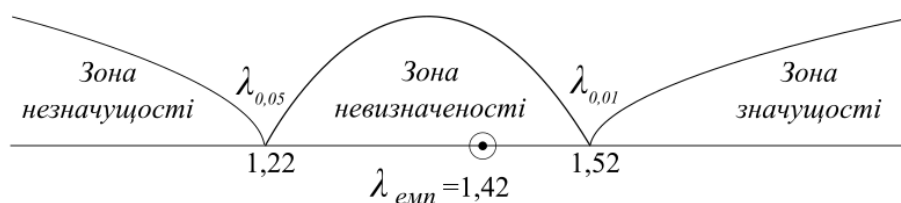


Рис. 4.28. Вісь значущості для емпіричного значення критерію типу Колмогорова-Смирнова для розподілів за результатами першого контрольного тестового опитування з теорії ОС

Для перевірки достовірності одержаного результату скористаємось ϕ -критерієм Фішера. Розподіл на підгрупи «є ефект» і «немає ефекту» подано у табл. 4.19.

Таблиця 4.19

Розподіл на підгрупи «є ефект» та «немає ефекту»

Групи	Емпіричні частоти вибору		Всього
	Експериментальна група	Контрольна група	
I група: достатній і високий рівні («є ефект»)	30	17	47
II група: початковий і середній рівні («немає ефекту»)	20	36	56
Всього	50	53	103

Будуємо чотириклітинну таблицю (табл. 4.20) з метою виявлення відмінностей між підгрупами «є ефект» та «немає ефекту» у КГ та ЕГ.

Таблиця 4.20

**Чотириклітинна таблиця для розрахунку ф-критерію Фішера
за результатами першого контрольного тестового опитування
з теорії операційних систем**

Група	«Є ефект»		«Немає ефекту»		Всього
	кількість	%	кількість	%	
Експериментальна	30	60,0%	20	40,0%	50
Контрольна	17	32,1%	36	67,9%	53

Сформулюємо статистичні гіпотези:

H_0 – розподіли за рівнями навчальних досягнень у КГ та ЕГ однакові;

H_1 – розподіли за рівнями навчальних досягнень у КГ та ЕГ відрізняються.

За поданими у [152, с. 330] таблицями визначаємо величини кута ϕ , що відповідають процентним часткам спостережень з підгрупи «є ефект» у КГ та ЕГ. Для ЕГ: $\phi_1(60,0\%)=1,772$. Для КГ: $\phi_2(32,1\%)=1,205$.

Обчислюємо емпіричне значення критерію ϕ^* за формулою 4.2:

$$\phi_{\text{емп}}^* = (1,772 - 1,205) \cdot \sqrt{\frac{50 \cdot 53}{50 + 53}} \approx 0,567 \cdot 5,072 \approx 2,88$$

За таблицею [152, с. 332] визначаємо, що одержане $\phi_{\text{емп}}^*$ відповідає рівню значущості $\rho < 0,01$ (уточнене значення, розраховане засобами мови R, становить 0,00571).

Критичні значення для кутового перетворення Фішера дорівнюють

$$\phi_{\text{кр}}^* = \begin{cases} 1,64 & (\rho \leq 0,05) \\ 2,31 & (\rho \leq 0,01) \end{cases}$$

Отримане значення $\phi_{\text{емп}}^*$ більше за відповідне критичне: $2,88 > 2,31$, а отже, одержані результати достовірні для рівня значущості 0,01. Побудуємо вісь значущості (рис. 4.29).



Рис. 4.29. Вісь значущості для емпіричного значення φ -критерію Фішера для розподілів за результатами першого контрольного тестового опитування з теорії ОС

Обчислене значення $\varphi_{емп}^*$ міститься у зоні значущості. Маємо право відхилити гіпотезу H_0 і прийняти гіпотезу H_1 : розподіли за рівнями навчальних досягнень у КГ та ЕГ відрізняються.

Одержані дані також було опрацьовано засобами мови R. Для даних табл. 4.20 виконано двосторонній тест Фішера (функція `fisher.test`). Одержано мале р-значення (p-value) $p\text{-value} = 0,00571$, високе співвідношенням шансів (odds ratio) $OR = 3,138996$ при довірчому інтервалі для рівня довіри 95% (95 percent confidence interval) від 1,31652 до 7,72506, який не містить одиниці, що підтверджує зроблені раніше висновки про відхилення гіпотези H_0 і прийняття гіпотези H_1 : розподіли за рівнями навчальних досягнень у КГ та ЕГ відрізняються. Повні програмні коди мовою R та отримані результати подано у додатку С.

Проаналізуємо результати **другого контрольного тестового опитування з теорії ОС**. Частоти розподілів за рівнями навчальних досягнень подано у таблиці 4.21 та на рис. 4.30.

Таблиця 4.21

**Розподіли за рівнями навчальних досягнень
за результатами другого контрольного тестового опитування
з теорії операційних систем**

Рівні навчальних досягнень	Контрольна група		Експериментальна група	
	Кількість, чол.	%	Кількість, чол.	%
Початковий рівень	9	16,98%	2	4,00%
Середній рівень	29	54,72%	19	38,00%
Достатній рівень	14	26,42%	20	40,00%
Високий рівень	1	1,89%	9	18,00%

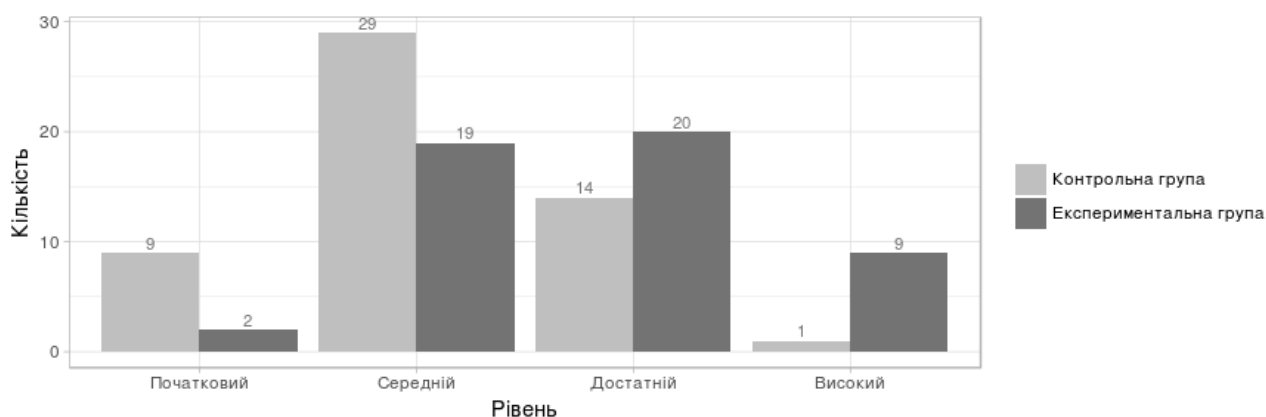


Рис. 4.30. Розподіли за рівнями навчальних досягнень на основі результатів другого контрольного тестового опитування з теорії ОС

Перевіримо достовірність даних експерименту за допомогою критерію типу Колмогорова-Смирнова (табл. 4.22).

Таблиця 4.22

Розрахунок λ -критерію у процесі зіставлення емпіричних розподілів за рівнями навчальних досягнень у контрольній та експериментальній групах

Рівні навчальних досягнень	Емпіричні частоти		Емпіричні відносні частоти		Накопичені емпіричні відносні частоти		Абсолютна різниця
	F_e	F_k	f_e	f_k	Σf_e	Σf_k	
Початковий	2	9	0,040	0,170	0,040	0,170	0,130
Середній	19	29	0,380	0,547	0,420	0,717	0,297
Достатній	20	14	0,400	0,264	0,820	0,981	0,161
Високий	9	1	0,180	0,019	1,000	1,000	0,000
Всього	50	53	1,000	1,000			

Максимальна різниця між двома відповідними накопиченими відносними частотами (останній стовпчик табл. 4.22) складає $d_{\max}=0,297$. Обчислимо значення λ за формулою 4.1:

$$\lambda = 0,297 \cdot \sqrt{\frac{50 \cdot 53}{50 + 53}} \approx 0,297 \cdot 5,072 \approx 1,506$$

Отримане критичне значення λ більше за критичне $\lambda_{0,05} = 1,224$, але менше за критичне $\lambda_{0,01} = 1,517$ (рис. 4.31).

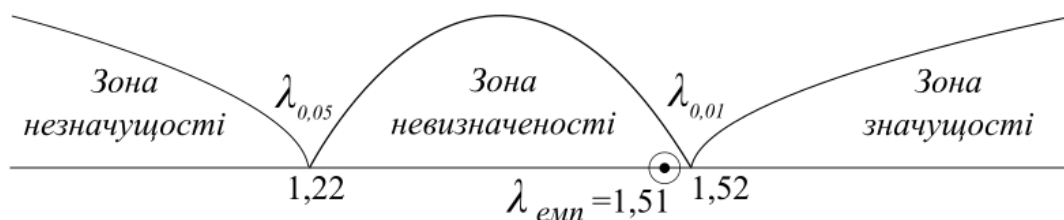


Рис. 4.31. Вісь значущості для емпіричного значення критерію типу Колмогорова-Смирнова для розподілів за результатами другого контрольного тестового опитування з теорії ОС

Для перевірки достовірності одержаного результату скористаємось кутовим перетворенням Фішера. Поділ аналізованих вибірок на підгрупи «є ефект» – «немає ефекту» наведено у табл. 4.23.

Таблиця 4.23

Розподіл на підгрупи «є ефект» та «немає ефекту»

Групи	Емпіричні частоти вибору		Всього
	Експериментальна група	Контрольна група	
I група: достатній і високий рівні («є ефект»)	29	15	44
II група: початковий і середній рівні («немає ефекту»)	21	38	59
Всього	50	53	103

Складемо чотириклітинну таблицю (табл. 4.24) для розрахунку ϕ -критерію Фішера для виявлення відмінностей між підгрупами «є ефект» та «немає ефекту» у КГ та ЕГ.

Таблиця 4.24

Чотириклітинна таблиця для розрахунку ϕ -критерію Фішера за результатами другого контрольного тестового опитування з теорії ОС

Група	«Є ефект»		«Немає ефекту»		Всього
	кількість	%	кількість	%	
Експериментальна	29	58,0%	21	42,0%	50
Контрольна	15	28,3%	38	71,7%	53

Сформулюємо статистичні гіпотези:

H_0 – розподіли за рівнями навчальних досягнень у КГ та ЕГ однакові;

H_1 – розподіли за рівнями навчальних досягнень у КГ та ЕГ відрізняються.

За таблицями, наведеними у [152, с. 330], визначаємо величини кута φ , що відповідають процентним часткам спостережень з підгрупи «е ефект» в КГ та ЕГ. Для ЕГ маємо $\varphi_1(58,0\%)=1,731$, для КГ – $\varphi_2(28,3\%)=1,122$.

Розраховуємо емпіричне значення критерію φ^* за формулою 4.2:

$$\varphi_{\text{емп}}^* = (1,731 - 1,122) \cdot \sqrt{\frac{50 \cdot 53}{50 + 53}} \approx 0,609 \cdot 5,072 \approx 3,09$$

За таблицею [152, с. 332] обчислюємо, що одержане $\varphi_{\text{емп}}^*$ відповідає рівню значущості $\rho < 0,001$ (уточнене значення, розраховане засобами мови R, становить 0,002893).

Критичні значення для кутового перетворення Фішера дорівнюють

$$\varphi_{\text{кр}}^* = \begin{cases} 1,64 (\rho \leq 0,05) \\ 2,31 (\rho \leq 0,01) \end{cases}$$

Значення $\varphi_{\text{емп}}^*$ більше за відповідне критичне: $3,09 > 2,31$, а отже, одержані результати достовірні для рівня значущості 0,01. Побудуємо вісь значущості (рис. 4.32).



Рис. 4.32. Вісь значущості для емпіричного значення φ -критерію Фішера для розподілів за результатами другого контрольного тестового опитування з теорії ОС

Обчислене значення $\varphi_{\text{емп}}^*$ міститься у зоні значущості. Маємо право відхилити гіпотезу H_0 і прийняти гіпотезу H_1 : розподіли за рівнями навчальних досягнень у КГ та ЕГ відрізняються.

Також було здійснено опрацювання одержаних даних засобами мови R. Для даних табл. 4.24 виконано двосторонній тест Фішера (функція `fisher.test`). Одержано мале р-значення $p\text{-value} = 0,002893$, високе співвідношенням шансів (odds ratio) $OR = 3,453334$ при довірчому інтервалі для рівня довіри 95% (95 percent confidence interval) від 1,430947 до 8,644717, який не містить одиниці,

що підтверджує зроблені раніше висновки про відхилення гіпотези H_0 і прийняття гіпотези H_1 : розподіли за рівнями навчальних досягнень у КГ та ЕГ відрізняються. Повні програмні коди мовою R, а також отримані результати подано у додатку С.

Таким чином, було одержано та порівняно розподіли за рівнями навчальних досягнень студентів у КГ та ЕГ: середній бал, одержаний за захист лабораторних робіт; бали за два контрольні тестові опитування з теорії ОС; бали за тестове опитування «Unix-подібні операційні системи». Порівняння показало наявність статистично значущих відмінностей між розподілами за рівнями навчальних досягнень у КГ та ЕГ. Враховуючи попередньо доведену однорідність вибірок, виявлені відмінності дають змогу підтвердити сформульовану у п. 2.1 *гіпотезу дослідження*: якщо навчання Unix-подібних ОС здійснювати на основі розробленої методики застосування технологій віртуалізації, то це дасть змогу покращити процес формування інформатичних компетентностей щодо ОС у бакалаврів інформатики.

Одержані результати дають підставу вважати запропоновану методику ефективною для організації навчання Unix-подібних ОС під час підготовки бакалаврів інформатики, а її впровадження у навчальний процес – доцільним.

Висновки до розділу 4

У розділі 4 описано організацію і проведення педагогічного експерименту, а також статистичне опрацювання одержаних результатів.

Педагогічний експеримент тривав з 2013 по 2018 рр. й відбувався у наступні етапи: *констатувальний етап* (2013-2015; аналіз вітчизняного та закордонного досвіду навчання ОС і ролі у ньому засобів віртуалізації; бесіди з вітчизняними студентами, викладачами курсу з ОС, фахівцями в галузі; аналіз власного досвіду викладання курсу з ОС бакалаврам інформатики; аналіз підсумкових балів майбутніх бакалаврів інформатики за курс з ОС та контрольне тестове опитування «Unix-подібні операційні системи»; анкетування викладачів щодо їхнього досвіду застосування технологій віртуалізації під час викладання курсу з ОС; розробка методики застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики; уточнення гіпотези дослідження; підбір експериментальної бази для формувального етапу експерименту, формування контрольної та експериментальної групи й перевірка однорідності вибірок); *формувальний етап* (2015-2018; реалізація навчання курсу з ОС відповідно до розробленої методики; організація і проведення контрольних заходів у КГ та ЕГ; аналіз одержаних експериментальних даних; узагальнення результатів аналізу експериментальних даних; розробка методичних рекомендацій із застосування технологій віртуалізації у навчанні ОС бакалаврів інформатики; уточнення процедури добору засобів віртуалізації для навчання операційних систем бакалаврів інформатики; експертне опитування з метою вибору змістового модуля для опису на його прикладі розробленої методики; впровадження результатів дослідження у процес підготовки бакалаврів інформатики у вітчизняних ЗВО).

У межах **констатувального етапу** експерименту було виявлено такі особливості курсу з ОС у вітчизняних та закордонних ЗВО, як: неоднаковість за обсягом годин та навчальними програмами; часте використання технологій віртуалізації під час виконання лабораторних робіт; застосування одного чи

декількох засобів віртуалізації під час виконання лабораторних робіт; пріоритетність вивчення Unix-подібних ОС (особливо у закордонних ЗВО) і можливість вивчення також іншої ОС (Windows, Android та ін.); навчання основ роботи із застосуванням ПЗ та ін. Відзначено наступні проблеми, пов'язані із застосуванням технологій віртуалізації під час навчання Unix-подібних ОС: апаратні та програмні збої; низька швидкість роботи віртуалізованої ОС; відсутність адаптованих інструкцій до віртуалізаційного ПЗ; прив'язаність лабораторних занять до конкретних аудиторій; неоднаковість умов здійснення самостійної роботи студентів з віртуалізованою ОС. Аналіз балів майбутніх бакалаврів інформатики Житомирського державного університету імені Івана Франка (2010-11, 2011-12, 2013-14 н.р.) та Дрогобицького державного педагогічного університету імені Івана Франка (II семестр 2011-12 н.р. – I семестр 2012-13 н.р.; II семестр 2012-13 н.р. – I семестр 2013-14 н.р.; 2013-14 н.р.) із загальним обсягом вибірки 98 осіб виявив, що середні бали за лабораторні роботи, присвячені ОС Linux, у проаналізованих випадках були загалом нижчі за підсумкові бали з дисципліни «Операційні системи та системне програмування» й лише у поодиноких випадках дорівнюють підсумковому балу або дещо перевищують його. На підставі одержаних даних, а також результатів бесід зі студентами та викладачами, у бакалаврів інформатики констатовано загалом нижчий рівень навчальних досягнень, пов'язаних із Unix-подібними ОС, порівняно з аналогічним рівнем для ОС Windows. Проведено анкетування викладачів ЗВО України, які викладали курс з ОС майбутнім бакалаврам інформатики та студентам інших дисциплін (загалом 9 респондентів). Результати анкетування виявили наявність низки відмінностей у навчанні ОС у вітчизняних ЗВО, які стосувалися зокрема: ОС, що вивчалися; категорій використовуваного віртуалізаційного ПЗ та конкретних засобів віртуалізації; технічних умов; співвідношення застосування під час лабораторних робіт графічного інтерфейсу та інтерфейсу командного рядка; оцінювання викладачами важливості низки факторів добору засобів віртуалізації та ін. Результати констатувального експерименту привели до

висновку про необхідність індивідуального добору засобів віртуалізації для курсу з ОС, про потребу залучення альтернативних засобів віртуалізації поряд з основними. Підготовлено експериментальний майданчик на базі Житомирського державного університету імені Івана Франка (2014-15, 2015-16 н. р.), Дрогобицького державного педагогічного університету імені Івана Франка (2014-15, 2015-16 н. р.), Сумського державного педагогічного університету імені А. С. Макаренка (2015-16 н. р.). Учасниками експерименту стали майбутні бакалаври інформатики (на момент проведення – студенти напряму підготовки 6.040302 «Інформатика»), обсяг вибірки – 103 особи (53 – КГ, 50 – ЕГ). Однорідність вибірок перевірено шляхом аналізу середнього балу, одержаного студентами на попередніх сесіях (критерій типу Колмогорова-Смирнова, емпіричне значення критерію 0,34, менше за критичне $\lambda_{0,05} = 1,22$), а також аналізу частки студентів, які мали досвід користування Unix-подібними ОС і досвід користування засобами віртуалізації до початку проходження курсу (кутове перетворення Фішера, емпіричні значення критерію 0,38 для частки студентів з досвідом користування Unix-подібними ОС і 0,57 для частки студентів з досвідом користування засобами віртуалізації, менші за критичне $\varphi_{0,05} = 1,64$). За результатами аналізу всіх трьох ознак прийнято гіпотезу про однаковість розподілу досліджуваних ознак у КГ та ЕГ на рівні значущості 0,05.

Формувальний етап експерименту проводився на базі обраного експериментального майданчика з 2015 по 2016 рік. З метою підтвердження чи спростування гіпотези про покращення процесу формування інформатичних компетентностей щодо ОС у бакалаврів інформатики у разі, якщо навчання Unix-подібних ОС здійснюватиметься на основі розробленої методики застосування технологій віртуалізації, було одержано та проаналізовано наступні бали студентів контрольної та експериментальної групи: бал за тестове опитування «Unix-подібні операційні системи», середній бал за захист лабораторних робіт, бали за два контрольні тестові опитування з теорії ОС. Застосовано критерій типу Колмогорова-Смирнова (емпіричні значення

критерію 1,43, 1,53, 1,42, 1,51 відповідно, більші за критичне $\lambda_{0,05} = 1,22$, але менші або несуттєво більші за критичне $\lambda_{0,01} = 1,52$). З метою уточнення результатів використано кутове перетворення Фішера (емпіричні значення критерію 2,90, 3,11, 2,88, 3,09 відповідно, більші за критичне $\varphi_{0,01} = 2,31$). В усіх чотирьох випадках маємо право прийняти гіпотезу про наявність статистично значущих відмінностей розподілу досліджуваних ознак у КГ та ЕГ на рівні значущості 0,01.

Отже, під час навчання Unix-подібних ОС на основі розробленої методики застосування технологій віртуалізації спостерігалось покращення процесу формування інформатичних компетентностей щодо ОС у бакалаврів інформатики. Це дає підставу вважати методику ефективною для організації навчання Unix-подібних ОС під час підготовки бакалаврів інформатики, а впровадження методики у навчальний процес – доцільним.

Основні результати проведеного у розділі 4 дослідження опубліковано у роботах [39; 45; 160].

ВИСНОВКИ

Відповідно до мети і завдань дисертаційного дослідження отримано наступні основні **результати**: з'ясовано стан розробленості проблеми застосування технологій віртуалізації Unix-подібних операційних систем у підготовці бакалаврів інформатики у практиці навчання операційних систем, уточнено понятійний апарат дослідження; проаналізовано наявні технології з метою добору засобів віртуалізації для навчання Unix-подібних операційних систем бакалаврів інформатики; визначено критерії показники, рівні сформованості інформатичних компетентностей бакалаврів інформатики щодо операційних систем; розроблено модель застосування технологій віртуалізації Unix-подібних операційних систем у підготовці бакалаврів інформатики; розроблено методику застосування технологій віртуалізації Unix-подібних операційних систем у підготовці бакалаврів інформатики та експериментально перевірено її педагогічну доцільність.

Одержані результати дослідження дають підстави зробити наступні висновки.

1. Віртуалізацію варто розуміти як поняття, що об'єднує технології, засоби, методи тощо, котрим притаманні три головні риси: (1) поділ ресурсів одного фізичного комп'ютера на декілька взаємно незалежних віртуальних середовищ або об'єднання ресурсів кількох фізичних комп'ютерів в одне віртуальне середовище; (2) оперативність переходу з одного віртуального середовища в інше; (3) приховування реальних фізичних ресурсів та заміна їх абстракціями.

Побудовано узагальнену систематизацію технологій віртуалізації за напрямом (віртуалізація серверів на основі віртуальних машин, віртуальних контейнерів; віртуалізація настільних ОС на основі віртуальних машин, віртуальних контейнерів; віртуалізація програмних застосунків; віртуалізація робочого столу на основі сервісів віддаленого робочого столу, інфраструктури віртуального робочого столу) та за методом (програмна віртуалізація за допомогою повної емуляції; емуляції системних бібліотек; квазіемуляції на

основі динамічної трансляції, паравіртуалізації; апаратна віртуалізація на основі технологій AMD-V, Intel VT-x).

В результаті аналізу вітчизняного та закордонного досвіду навчання ОС у ЗВО встановлено, що важливим є індивідуальний добір засобів віртуалізації для курсу з ОС, необхідним є залучення альтернативних засобів віртуалізації поряд з основними.

2. Добір програмних засобів віртуалізації у навчанні Unix-подібних ОС бакалаврів інформатики доцільно здійснювати на основі сформульованих критеріїв та показників: відповідність використовуваної технології віртуалізації потребам курсу (можливість формування інформатичних компетентностей бакалаврів інформатики щодо ОС; швидкість; надійність; універсальність щодо гостей ОС; універсальність комп'ютерного класу; м'якість переходу для викладача та навчального закладу; уникнення надання студенту прав адміністратора; простота організації студентської самостійної роботи; оперативність переключення між ОС; простота вилучення/заміни ОС, що вивчається), універсальність (універсальність щодо основних ОС; універсальність щодо гостей ОС; універсальність щодо технологій апаратної віртуалізації), ліцензійна чистота (вид ліцензії), функціональність інтерфейсу (наявність графічного інтерфейсу; можливість керування з командного рядка; наявність україномовного інтерфейсу; можливість віддаленого доступу), популярність та підтримка (середній пошуковий інтерес; засоби підтримки користувачів; дата випуску останнього стабільного релізу). Водночас, встановлено суттєву залежність результату оцінювання програмних засобів за пропонованими критеріями і показниками від ваги показників і, як наслідок, потребу індивідуального добору засобів віртуалізації у кожному окремому випадку.

3. Виокремлено інформатичні компетентності бакалаврів інформатики щодо ОС: знати та розуміти основні етапи, напрями та тенденції розвитку ОС; орієнтуватися у класифікації ОС, з урахуванням їх архітектури та сфери застосування; знати та розуміти теоретичні основи будови та функціонування ОС;

вміти використовувати інтерфейс користувача Unix-подібних ОС (графічний інтерфейс і/або інтерфейс командного рядка); вміти налаштовувати мережне з'єднання у Unix-подібних ОС; вміти встановлювати програмне забезпечення у Unix-подібних ОС, виконувати його наступне оновлення та деінсталяцію; вміти відстежувати роботу процесів та потоків в ОС і керувати ними; вміти здійснювати моніторинг використання різних видів пам'яті; володіти основами роботи з файловою системою у Unix-подібних ОС; володіти основами роботи з налаштуваннями системи безпеки ОС; володіти основами системного програмування мовою C++ у Linux і / або у Windows; вміти одержувати довідкові відомості щодо використання ОС та програмування у них.

Діагностування рівня сформованості інформатичних компетентностей бакалаврів інформатики щодо ОС доцільно здійснювати на основі розроблених критеріїв (теоретичний, адміністративний, програмістський, самонавчальний) та показників за чотирма рівнями сформованості інформатичних компетентностей бакалаврів інформатики щодо ОС: початковий, середній, достатній, високий.

4. Модель застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики складається з цільового, змістового, технологічного, діагностичного й аналітичного компонентів. В основу моделі покладено послідовність розроблення методики, враховуючи послідовне вдосконалення з кожним проведенням курсу з ОС.

5. Застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики доцільно здійснювати за розробленою методикою, в основу якої покладено варіативний підхід до використання таких технологій. Підхід полягає у поєднанні кількох засобів віртуалізації у межах одного курсу з ОС задля відповідності індивідуальним особливостям проведення курсу, індивідуальним потребам студентів і забезпечення стійкості у разі програмних та апаратних збоїв. Пропонований підхід передбачає ознайомлення викладачів та студентів з різноманітністю засобів віртуалізації, наведення прикладу поєднання кількох засобів віртуалізації у межах одного курсу з ОС,

адаптування навчально-методичних матеріалів до використання різних засобів віртуалізації. Розроблена методика містить мету, завдання, зміст, форми, методи і засоби. Організацію навчального процесу відповідно до пропонованої методики варто здійснювати за методичним посібником з дисципліни «Операційні системи та системне програмування» для навчання майбутніх бакалаврів інформатики та методичними рекомендаціями щодо використання технологій віртуалізації у навчанні ОС бакалаврів інформатики. Для врахування індивідуальних особливостей проведення курсу з ОС добір засобів віртуалізації доцільно здійснювати на основі пропонованої процедури добору цих засобів.

За результатами експериментальної перевірки методики підтверджено гіпотезу дослідження і встановлено її педагогічну доцільність для організації навчання Unix-подібних ОС бакалаврів інформатики. Впровадження розробленої методики у навчальний процес дає змогу підвищити результативність формування інформатичних компетентностей щодо ОС у бакалаврів інформатики.

Виконане дослідження не вичерпує всіх аспектів окресленої проблеми. Подальші наукові розвідки доцільно спрямувати на обґрунтування та розроблення методичної системи застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики у ЗВО, науково-методичний пошук можливостей використання таких технологій в освіті дорослих.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Г. Л. Абдулгалимов, Система профессиональной подготовки преподавателя информатики: компетентностный подход, проектирование, внедрение: монография, М.: РИЦ МГГУ им. М. А. Шолохова, 2008.
2. О. Б. Авраменко, О. В. Жмуд, "Зміст та структура предметної компетентності з архітектури комп'ютера та конфігурації комп'ютерних систем майбутніх учителів інформатики", *Вісник Житомирського державного університету імені Івана Франка*, № 3 (89), с. 17-21, 2017.
3. Г. Атанов, "Деятельностный подход в обучении", *Educational Technology & Society*, № 4 (4), с. 48-55, 2001. [Електронний ресурс]. Доступно: <http://cyberleninka.ru/article/n/deyatelnostnyy-podhod-vobuchenii>.
4. А. Батырь, "Виртуализация: технологические подходы", *PC Magazine*, 2009. [Электронный ресурс] Доступно: <http://www.pcmag.ru/solutions/detail.php?ID=34643>. Дата обращения: Июнь 22, 1018.
5. А. Є. Батюк, Д. Є. Ванькевич, Г. Г. Злобін, "Використання технологій віртуалізації в спецкурсі «Системне адміністрування ОС Linux»", *Електроніка та інформаційні технології*, № 3, с. 220-225, 2013.
6. В. С. Безрукова, *Педагогика. Проективная педагогика: учеб. пособ. для инженерно-пед. институтов и индустриально-пед. техникумов*. Екатеринбург: Деловая книга, 1996.
7. В. Ю. Биков, "Ключові чинники та сучасні інструменти розвитку системи освіти", *Інформаційні технології і засоби навчання*, № 2, 2007.
8. В. Ю. Биков, *Моделі організаційних систем відкритої освіти: Монографія*. К.: Атіка, 2009.
9. В. Ю. Биков, "Навчальне середовище сучасних педагогічних систем", *Професійна освіта: педагогіка і психологія: Україно-польський журнал*, вид. IV, с. 42-70, 2004.
10. В. Ю. Биков, "Основні принципи відкритої освіти", *Педагогічні і психологічні науки в Україні. Збірник наукових праць до 15-річчя АПН України у 5 томах*, т. 2, с. 67-81, 2007.

11. В. Ю. Биков, "Теоретико-методологічні засади моделювання навчального середовища сучасних педагогічних систем", у *Інформаційні технології і засоби навчання: Зб. наук. праць*, В. Ю. Биков і Ю. О. Жук, Ред. К.: Атіка, 2005, с. 5-15.
12. В. Ю. Биков, "Технології хмарних обчислень, ІКТ-аутсорсінг та нові функції ІКТ-підрозділів навчальних закладів і наукових установ", *Інформаційні технології в освіті: Зб. наук. праць*, №10, с. 8-23, 2011.
13. В. Ю. Биков, І. М. Ромашко, "Освітні системи із забезпечення рівного доступу до якісної освіти впродовж життя", *Інформаційні технології і засоби навчання*, № 48, 2008.
14. В. Ю. Биков, та В. Д. Руденко, *Системи управління інформаційними базами даних в освіті: Навч. посібник для студ. вищ. навч. закладів*. К.: ІЗМН, 1996.
15. М. Ф. Бондаренко, О. Г. Качко, *Операційні системи: навч. посіб.*. Харків: Компанія СМІТ, 2008.
16. А. А. Братанич, "Особистісно орієнтована технологія інформатичної підготовки майбутніх учителів", *Науковий часопис Національного педагогічного університету імені М. П. Драгоманова. Серія 5: Педагогічні науки: реалії та перспективи*, № 51, с. 41-45, 2015.
17. Э. Брауде, *Технология разработки программного обеспечения*, СПб.: Питер, 2004.
18. Л. В. Брескіна, "Професійна підготовка майбутніх вчителів інформатики на основі сучасних мережевих інформаційних технологій", автореф. дис. канд. пед. наук, Нац. пед. ун-т імені М.П.Драгоманова, К., 2003.
19. В. Брязкун, "До уточнення поняття «віртуалізація» як філософської категорії", *Наукові записки Національного університету "Острозька академія", Серія: Філософія*, № 8, с. 93-103, 2011.
20. Т. А. Вакалюк, "Структурно-функціональна модель хмарно орієнтованого навчального середовища для підготовки бакалаврів інформатики", *Інформаційні технології і засоби навчання*, т. 59, №3, с. 51-61, 2017.

21. Т. Я. Вдовичин, "Навчання бакалаврів інформатики з використанням мережних технологій відкритих систем у педагогічному університеті", *Інформаційні технології і засоби навчання*, т. 58, №2, с. 169-181, 2017.
22. Верховна Рада України. VII скликання, 4 сесія. (2014, Лип. 1, ред. від 2018, Січ. 1). *Закон №1556-VII, Про вищу освіту*. [Електронний ресурс]. Доступно: <http://zakon5.rada.gov.ua/laws/show/1556-18>.
23. "Виртуальная машина", Microsoft Press. Толковый словарь по вычислительной технике. Издательский отдел "Русская редакция" ТОО "Channel Trading Ltd", М., 1995.
24. О. Є. Висоцька, "Віртуалізація соціального у постмодерному суспільстві: від раціональної до постраціональної комунікації", *Вісник Черкаського університету*, № 154, с. 5-14, 2009.
25. "ВНТУ", *Офіційний сайт Вінницького національного технічного університету*. [Електронний ресурс]. Доступно: <http://vntu.edu.ua/>. Дата звернення: Січ. 31, 2019.
26. С. В. Воеводін, "Віртуальні вимірювальні прилади у лабораторному практикумі з твердотільної електроніки", *Комп'ютер у школі та сім'ї*, № 1, с. 31-35, 2016.
27. В. М. Волохов, *Методичні рекомендації до лабораторного практикуму побудови мовних процесорів з дисципліни «Системне програмування»*, К.: 2013.
28. В. Ю. Габрусев, "Зміст і методика вивчення шкільного курсу інформатики на основі вільно поширюваної операційної системи LINUX", дис. канд. пед. наук, НПУ ім. М.П.Драгоманова, К., 2003.
29. В. Ю. Габрусев, В. В. Лапінський, О. В. Нестеренко, *Основи операційних систем: ядро, процес, потік: навч. посібник*. Тернопіль: «Навчальна книга – Богдан», 2007.
30. Д. В. Галкин, "Проблемы образования в контексте информатизации: в поисках модели критической педагогики", *Гуманитарная информатика*, № 3, 2012.

31. М. Д. Гінзбург, "Application. Як це буде українською?", *Інформаційно-керуючі системи на залізничному транспорті*, №3, с. 81-84, 2004.
32. О. Г. Глазунова, О. В. Якобчук, "Проектування архітектури хмарно-орієнтованого інформаційно-освітнього середовища для підготовки майбутніх фахівців з інформаційних технологій", *Інформаційні технології та засоби навчання*, т. 44, № 6, с. 141-156, 2014.
33. Я. М. Глинський, В. А. Ряжська, *Linux-практикум з інформатики*, Львів: СПД Глинський, 2004.
34. "Головна – Донецький національний університет імені Василя Стуса", *Офіційний сайт*. [Електронний ресурс]. Доступно: <https://www.donnu.edu.ua/uk/>. Дата звернення: Січ. 31, 2019.
35. "Головна – ХАІ", *Офіційний сайт Національного аерокосмічного університету ім. М. Є. Жуковського*. [Електронний ресурс]. Доступно: <https://khai.edu.ua/>. Дата звернення: Січ. 31, 2019.
36. "Головна сторінка – Харківський національний економічний університет імені Семена Кузнеця (ХНЕУ ім. С. Кузнеця)", *Офіційний сайт*. [Електронний ресурс]. Доступно: <https://www.hneu.edu.ua/>. Дата звернення: Січ. 31, 2019.
37. О. С. Головня, "Аналіз досвіду навчання операційних систем у зарубіжних вищих освітніх закладах", на *Всеукраїнській науково-практичній конференції з міжнародною участю "Сучасні інформаційні технології в освіті та науці"*, Житомир, 10-11 листопада, 2016, с. 194-199.
38. О. С. Головня, "Аналіз зарубіжного досвіду навчання операційних систем у вищій школі", *Комп'ютер у школі та сім'ї*, № 1, с. 19-23, 2017.
39. О. С. Головня, "Варіативний підхід до застосування засобів віртуалізації Unix-подібних операційних систем у підготовці бакалаврів інформатики", *Науковий часопис НПУ імені М. П. Драгоманова. Серія 2: Комп'ютерно-орієнтовані системи навчання*, №19(16), с. 228-233, 2017.
40. О. С. Головня, "Критерії добору віртуалізаційного ПЗ у навчанні Unix-

- подібних операційних систем", на *IX міжнародній конференції ІТЕА-2014*, Київ, 26 листопада, 2014, с. 124-130.
41. О. С. Головня, "Критерії добору програмних засобів віртуалізації Unix-подібних операційних систем у підготовці бакалаврів інформатики", на *II всеукраїнській науково-практичній конференції молодих науковців "Наукова молодь – 2014"*, Київ, 11 грудня, 2014, с. 101-104.
 42. О. С. Головня, "Критерії добору програмних засобів віртуалізації у навчанні Unix-подібних операційних систем", *Інформаційні технології в освіті*, № 24, с. 119-133, 2015.
 43. О. С. Головня, *Операційні системи та системне програмування: Методичний посібник для студ. вищ. навч. закл..* Житомир: Вид-во ЖДУ ім. І. Франка, 2013.
 44. О. С. Головня, *Операційні системи та системне програмування: Методичний посібник для студ. вищих навч. закл..* 2-ге вид., переробл. і доповн.. Житомир: Рута, 2016.
 45. О. С. Головня, "Поєднання кількох засобів віртуалізації у навчанні бакалаврів інформатики дисципліни «Операційні системи та системне програмування»", на *VII міжнародній науково-технічній конференції "Інформаційно-комп'ютерні технології – 2016"*, Житомир, 22-23 квітня, 2016, с. 220-221.
 46. О. С. Головня, "Психолого-педагогічні особливості студентів напряму підготовки «Інформатика» в контексті навчання операційних систем", на *науково-практичній семінар "Міжнародна співпраця у просторі відкритої освіти"*, Київ, 5 листопада, 2012.
 47. О. С. Головня, "Розробка методичних рекомендацій із застосування технологій віртуалізації у курсі з операційних систем", на *II всеукраїнській науково-практичній конференції з міжнародною участю "Сучасні інформаційні технології в освіті та науці"*, присвяченій 10-й річниці функціонування Інтернет-порталу E-OLYMP, Житомир, 9-10 листопада, 2017, с. 119-123.

48. О. С. Головня, "Систематизація технологій віртуалізації", *Інформаційні технології в освіті*, № 12, с. 127-133, 2012.
49. О. С. Головня, "Систематизація технологій віртуалізації", на звітній науковій конференції Інституту інформаційних технологій і засобів навчання НАПН України, Київ, 29 березня, 2012, с. 87-88.
50. О. С. Головня, *Технології віртуалізації у навчанні операційних систем бакалаврів інформатики: Методичні рекомендації для викладачів вищ. навч. закл.*. Житомир: Рута, 2017.
51. О. С. Головня, "Хмарні інтегровані середовища розробки у курсі з операційних систем для студентів педагогічних спеціальностей: досвід і перспективи", на III всеукраїнській науково-практичній конференції з міжнародною участю "Сучасні інформаційні технології в освіті та науці", Житомир, 8-9 листопада, 2018, с. 327-332.
52. С. У. Гончаренко, "Експеримент психолого-педагогічний", *Енциклопедія освіти*, В. Г. Кремін, Ред. К.: Юрінком Інтер, 2004, с. 253-255.
53. О. П. Горбачевська, Г. С. Драган, С. В. Єгоров, В. В. Лапінський, та Д. В. Сподарець, "Хмарні та грид-технології у навчальному процесі університетів", *Вища освіта України*, т. III, с. 143-153, 2014.
54. А. К. Гультяев, *Виртуальные машины: несколько компьютеров в одном*, СПб.: Питер, 2006.
55. А. М. Гуржій, Р. С. Гуревич, Л. Л. Коношевський, та О. Л. Коношевський, *Мультимедійні технології та засоби навчання: навчальний посібник*. Вінниця: Нілан-ЛТД, 2017.
56. Д. К. Джонс, *Методы проектирования*. М.: Мир, 1986.
57. Я. Дитрих, *Проектирование и конструирование: Системный подход*. М.: Мир, 1981.
58. "Дніпропетровський національний університет імені Олеся Гончара", *Офіційний сайт*. [Електронний ресурс]. Доступно: <http://www.dnu.dp.ua/>. Дата звернення: Січ. 31, 2019.
59. А. В. Добридень, "Навчання впродовж життя в контексті Болонського

- процесу", *Вища освіта України*, т. 1(8), дод. 3, с. 90-97, 2008.
60. В. В. Докучаєва, "Теоретико-методологічні основи проектування інноваційних педагогічних систем", дис. доктора пед. наук, Луганськ, 2007.
 61. В. Дьомкін, "Курс «Операційні системи» – «Операционные системы»", [Електронний ресурс]. Доступно: <http://vseloved.github.io/spos.html>. Дата звернення: Черв. 26, 2018.
 62. Н. Елманова, С. Пахомов, "Виртуальные машины", *КомпьютерПресс*, №9, с. 29-42, 2007.
 63. М. І. Жалдак, "Проблеми фундаменталізації змісту навчання інформатичних дисциплін в педагогічних університетах", *Науковий часопис Національного педагогічного університету імені М. П. Драгоманова. Серія 2: Комп'ютерно-орієнтовані системи навчання*, №17(24), с. 3-15, 2015.
 64. М. І. Жалдак, "Система підготовки вчителя до використання інформаційно-комунікаційних технологій в навчальному процесі", *Науковий часопис НПУ імені М. П. Драгоманова. Серія 2: Комп'ютерно-орієнтовані системи навчання*, №. 11, с. 3-15, 2011.
 65. М. І. Жалдак, В. В. Лапінський, М. І. Шут, "Комп'ютерно-орієнтовані засоби навчання математики, фізики, інформатики", *Інформатика*, № 3-4, с. 3-96, 2006.
 66. М. І. Жалдак, Ю. С. Рамський, М. В. Рафальська, "Модель системи соціально-професійних компетентностей вчителя інформатики", *Науковий часопис НПУ імені М.П.Драгоманова. Серія №2. Комп'ютерно-орієнтовані системи навчання*, № 7(14), с. 3-18, 2009.
 67. О. В. Жмуд, "Формування предметної компетентності з архітектури комп'ютера та конфігурації комп'ютерних систем у майбутніх учителів інформатики", автореферат дис. канд. пед. наук, НПУ імені М. П. Драгоманова, К., 2015.

68. Е. С. Заир-Бек, "Теоретические основы обучения педагогическому проектированию", дис. доктора пед. наук, СПб., 1995.
69. С. С. Занюк, *Психологія мотивації: навч. посібник*, К.: Либідь, 2002.
70. М. Г. Зубков, *Сучасний англо-український україно-англійський словник*, Х.: 2008.
71. С. Ильин, "VirtualBox Tips'n'Tricks: Неочевидные трюки использования виртуальной машины", *Хакер*, №12/10, 2011. [Электронный ресурс]. Доступно: <http://www.haker.ru/post/55217/>. Дата обращения: Июнь 22, 2018.
72. Л. Б. Ительсон, "Психологические теории научения и модели процесса обучения", *Советская педагогика*, № 3, с. 83-95, 1973.
73. Кабінет Міністрів України (Квіт. 29, 2015, ред. від Лют. 11, 2017) *Постанова, Про затвердження переліку галузей знань і спеціальностей, за якими здійснюється підготовка здобувачів вищої освіти*. [Електронний ресурс].
74. В. А. Кавалеров, "Концепт віртуалізації освіти в комунікаціях постмодерного суспільства", *Гуманітарний вісник Запорізької державної інженерної академії*, № 50, с. 94-100, 2012.
75. В. Е. Карпов, К. А. Коньков, *Основы операционных систем. Курс лекций. Учебное пособие*. М.: ИНТУИТ.РУ "Интернет-Университет Информационных Технологий", 2005.
76. О. М. Коберник, *Проектування навчально-виховного процесу в школі*. К.: Хрещатик, 1995.
77. Т. П. Кобильник, "Фундаментальність інформатичної освіти", *Науковий часопис НПУ ім. М.П. Драгоманова. Серія No2: Комп'ютерно-орієнтовані системи навчання*, № 5(12), с. 78-81, 2007.
78. У. П. Когут, "Класифікація та критерії вибору програмних засобів для фундаменталізації підготовки бакалаврів інформатики з інформатичних дисциплін", *Інформаційні технології в освіті*, № 11, с. 88-97, 2012.
79. І. І. Коновальчук, "Формування у майбутніх учителів умінь проектувати

- виховну діяльність: автореф. дис. канд. пед. наук, НПУ імені М. П. Драгоманова, К., 1999.
80. Л. М. Корчагіна, "Навчання впродовж життя як пріоритетний напрямок модернізації управління освітою в Україні", *Теорія та практика державного управління*, № 3, с. 326-331, 2008.
 81. "КПІ ім. Ігоря Сікорського – Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»", *Офіційний сайт*. [Електронний ресурс]. Доступно: <https://kpi.ua/>. Дата звернення: Січ. 31, 2019.
 82. «Криворізький державний педагогічний університет – КДПУ», *Офіційний сайт*. [Електронний ресурс]. Доступно: <https://kdpu.edu.ua/>. Дата звернення: Лют. 6, 2019.
 83. А. П. Кузьменко, Н. Б. Єпик, В. М. Кузьменко, "Математичні компетенції у підготовці бакалавра інформатики", *Психолого-педагогічні основи гуманізації навчально-виховного процесу в школі та ВНЗ*, № 1, с. 47-53, 2013.
 84. Г. П. Лаврентьєва, М. П. Шишкіна, *Методичні рекомендації з організації та проведення науково-педагогічного експерименту*, К.: ІТЗН, 2007.
 85. В. В. Лапінський, В. Ю. Габрусєв, Н. Я. Бачинська, *Основи операційних систем*, Тернопіль: «Навчальна книга – Богдан», 2002.
 86. В. В. Лаптев, Н. Н. Рыжова, М. В. Швецкий, *Методическая теория обучения информатике. Аспекты фундаментальной подготовки*, СПб.: Изд-во С.-Петербур. ун-та, 2003.
 87. М. П. Лапчик, "Структура и методическая система подготовки кадров информатизации школы в педагогических вузах", дис. доктора пед. наук, Омский гос. пед. ун-т., М., 1999.
 88. С. Г. Литвинова, "Теоретико-методичні основи проектування хмаро орієнтованого навчального середовища загальноосвітнього навчального закладу", дис. доктора пед. наук, Інститут ІТЗН, К., 2016.
 89. А. О. Лігоцький, *Теоретичні основи проектування сучасних освітніх*

систем. К.: Техніка, 1997

90. "Львівський національний університет імені Івана Франка", *Офіційний сайт*. [Електронний ресурс]. Доступно: <http://www.lnu.edu.ua/>. Дата звернення: Січ. 31, 2019.
91. З. Ф. Мазур, "Научно-педагогические основы проектирования средств и технологий интеллектуальной собственности в сфере образования", дис. доктора пед. наук, М., 1998.
92. Г. Майерс, *Надежность программного обеспечения*, М.: Мир, 1980.
93. І. Д. Малицька, "Віртуальні спільноти як інноваційні освітні середовища в системах освіти зарубіжних країн", *Інформаційні технології в освіті*, № 15, с. 276-283, 2013.
94. Ю. Г. Мартыненко, "Применение новых информационных технологий в преподавании фундаментальных наук", *Соросовский образовательный журнал*, № 3, с. 130-138, 1997.
95. Е. И. Машбиц, *Компьютеризация обучения: Проблемы и перспективы*. М.: Знание, 1986.
96. Ю. Медведев, "Что такое виртуализация", *PC Magazine*, 2007 [Электронный ресурс] Доступно: <http://www.pcmag.ru/reviews/detail.php?ID=9291>. Дата обращения: Июнь 22, 1018.
97. О. В. Мерзликін, "Хмарні технології як засіб формування дослідницьких компетентностей старшокласників у процесі профільного навчання фізики", дис. канд. пед. наук, Інститут ІТЗН, К., 2016.
98. Міністерство освіти і науки України (2010). Галузевий стандарт вищої освіти України. Освітньо-кваліфікаційна характеристика бакалавр. Галузь знань 0403 Системні науки та кібернетика. Напрямок підготовки 040302 Інформатика. Київ.
99. Міністерство освіти і науки України (2010). Галузевий стандарт вищої освіти України. Освітньо-професійна програма підготовки бакалавр. Галузь знань 0403 Системні науки та кібернетика. Напрямок підготовки 040302 Інформатика. Київ.

100. Міністерство освіти і науки України. (2017) Стандарт вищої освіти України першого (бакалаврського) рівня освіти, ступеня вищої освіти бакалавр (проект). Галузь знань 01 Освіта. Спеціальність 014 Середня освіта. Предметна спеціалізація 014.09 Середня освіта (Інформатика). Київ.
101. Н. Є. Мойсеюк, *Педагогіка. Навчальний посібник*, 5-е вид., доповн. і переробл., К., 2007.
102. В. М. Монахов, "Проектирование и внедрение новых информационных технологий обучения", *Советская педагогика*, № 7, с. 17-22, 1990.
103. В. М. Монахов, *Технологические основы проектирования и конструирования ученого процесса: монографія*. Волгоград: Перемена, 1995.
104. Н. М. Морзе, "Система методичної підготовки майбутніх вчителів інформатики в педагогічних університетах", дис. доктора пед. наук, Національний педагогічний ун-т ім. М. П. Драгоманова, К., 2003.
105. Г. Е. Муравьёва, "Вопросы теории проектирования образовательных процессов", *Педагогическое образование и наука*, № 4, с. 14-21, 2002.
106. Нац. акад. пед. наук України, "Національна доповідь про стан і перспективи розвитку освіти в Україні", Педагогічна думка, Київ, 2016.
107. "Національний авіаційний університет", *Офіційний сайт*. [Електронний ресурс]. Доступно: <https://nau.edu.ua/>. Дата звернення: Січ. 31, 2019.
108. "Національний педагогічний університет імені М. П. Драгоманова", *Офіційний сайт*. [Електронний ресурс]. Доступно: <http://www.npu.edu.ua/>. Дата звернення: Січ. 31, 2019.
109. "Національний технічний університет – Харківський політехнічний інститут", *Офіційний сайт*. [Електронний ресурс]. Доступно: <http://www.kpi.kharkov.ua/ukr/>. Дата звернення: Січ. 31, 2019.
110. "Національний університет біоресурсів і природокористування України", *Офіційний сайт*. [Електронний ресурс]. Доступно: <https://nubip.edu.ua/>. Дата звернення: Січ. 31, 2019.

111. "Національний університет «Києво-Могилянська академія»", *Офіційний сайт*. [Електронний ресурс]. Доступно: <https://www.ukma.edu.ua/>. Дата звернення: Січ. 31, 2019.
112. "Національний університет «Львівська політехніка»", *Офіційний сайт*. [Електронний ресурс]. Доступно: <http://lp.edu.ua/>. Дата звернення: Січ. 31, 2019.
113. *Новые ценности образования. Тезаурус для учителей и школьных психологов*, М., 1995.
114. Ю. Г. Носенко, "Хмарні технології у просторі відкритої освіти", у *Моделювання й інтеграція сервісів хмаро орієнтованого навчального середовища: монографія*, С. Г. Литвинова, Ред. К.: ЦП «Компринт», 2015, с. 24-34.
115. "НТУ «Дніпровська політехніка»", *Офіційний сайт Національного технічного університету "Дніпровська політехніка"*. [Електронний ресурс]. Доступно: <http://www.nmu.org.ua/ua/>. Дата звернення: Січ. 31, 2019.
116. П. И. Образцов, А. И. Ахулкова, О. Ф. Черниченко, *Проектирование и конструирование профессионально-ориентированной технологии обучения: Учебно-методическое пособие*, Орел: ОГУ, 2003.
117. "Одеський національний політехнічний університет", *Офіційний сайт*. [Електронний ресурс]. Доступно: <http://opi.ua/>. Дата звернення: Січ. 31, 2019.
118. В. П. Олексюк, "Застосування віртуальних хмарних лабораторій у процесі підготовки майбутніх учителів інформатики", *Науковий часопис НПУ імені М. П. Драгоманова. Серія 2: Комп'ютерно-орієнтовані системи навчання*, № 15, с. 76-81, 2015.
119. "Операційні системи", *Інформація про курс на Moodle ЧНУ (Чорноморський національний університет імені Петра Могили)*. [Електронний ресурс]. Доступно: <http://moodle3.chmnu.edu.ua/course/info.php?id=723>. Дата звернення: Січ. 31, 2019.

120. *Операційні системи: Методичні рекомендації до лабораторних занять з навчальної дисципліни підготовки бакалаврів зі спеціальності*
121 *"Інженерія програмного забезпечення".* Х.: ХНЕУ ім. С. Кузнеця, 2016.
121. К. П. Осадча, "Засоби формування професійної компетентності майбутніх учителів інформатики", *Інформаційні технології і засоби навчання*, № 3(17), 2010.
122. Т. В. Отрошко, "Модель технічної компетентності майбутніх вчителів інформатики", *Проблеми інженерно-педагогічної освіти*, № 24-25, с. 177-188, 2009.
123. Т. В. Отрошко, "Система оцінювання технічної компетентності майбутніх учителів інформатики в процесі навчання комп'ютерних дисциплін", автореф. дис. канд. пед. наук, Українська інженерно-педагогічна академія, Харків, 2010.
124. "Офіційний сайт Тернопільського національного педагогічного університету імені Володимира Гнатюка". [Електронний ресурс]. Доступно: <http://tnpu.edu.ua/>. Дата звернення: Січ. 31, 2019.
125. М. П. Павленко, та Л. В. Павленко, "Використання технологій віртуалізації для навчання інженерів-педагогів IP-телефонії в комп'ютерних мережах", *Науковий вісник Мелітопольського державного педагогічного університету*, № 1(14), с. 269-274, 2015.
126. Н. С. Павлова, І. С. Войтович, "Особистісно орієнтований підхід як основа формування професійних компетентностей у майбутніх учителів інформатики", *Міжвузівський збірник "Комп'ютерно-інтегровані технології: освіта, наука, виробництво"*, № 4, с. 137-144, 2011.
127. Т. В. Панченко, *Методичні рекомендації до лабораторних занять з курсу «Системне програмування та операційні системи»*. К., 2009.
128. Х. В. Паньків, *Операційні системи: лабораторний практикум*. Івано-Франківськ: ІФНТУНГ, 2015.
129. Є. А. Пінчук, "Концептуальні основи та тенденції розвитку безперервної

- освіти", *Вісник НТУ «КПІ». Філософія. Психологія. Педагогіка*, № 1, с. 71-77, 2011.
130. "План лабораторних занять з дисципліни «Операційні системи». *Офіційний сайт Одеського національного політехнічного університету*. [Електронний ресурс]. Доступно: <http://opu.ua/upload/files/ppOS126B1.pdf>. Дата звернення: Січ. 31, 2019.
 131. Т. Ю. Подобедова, "Теория и практика педагогического проектирования", *Проблеми сучасної педагогічної освіти. Серія «Педагогіка і психологія»*, № 6, ч. 2, с. 81-87, 2004.
 132. О. І. Пометун, "Формування громадянської компетентності: погляд з позиції сучасної педагогічної науки", *Вісник програм шкільних обмінів*, № 23, с. 18-24, 2005.
 133. М. В. Попель, "Хмарний сервіс SageMathCloud як засіб формування професійних компетентностей вчителя математики", дис. канд. пед. наук, Інститут ІТЗН, К., 2017.
 134. "Про систему хмарних технологій та їх впровадження в НГУ", *Новини Національного технічного університету "Дніпровська політехніка"* (21.09.2016). [Електронний ресурс]. Доступно: http://www.nmu.org.ua/ua/content/news/?ELEMENT_ID=19884. Дата звернення: Січ. 31, 2019.
 135. "Проект «Хмарні технології»", *Кафедра комп'ютерних систем та мереж*, Національний аерокосмічний університет ім. М. Є. Жуковського, 2014.
 136. С. М. Прийма, "Формування технологічної культури майбутніх учителів інформатики у процесі професійно-педагогічної підготовки", дис. канд. пед. наук, Харківський національний педагогічний ун-т ім. Г. С. Сковороди, Х., 2006.
 137. О. Г. Прикот, "Педагогическое проектирование как рабочий инструмент методической службы школы", *Методист: науч.-метод. журнал*, № 2, с. 44-47, 2002.

138. Е. М. Пройдаков, Л. А. Теплицький, "Віртуальна машина", Англо-український тлумачний словник з обчислювальної техніки, Інтернету і програмування. Видавничий дім "СофтПрес", К., 2005.
139. *Професійна педагогічна освіта: особистісно орієнтований підхід: монографія*, О. А. Дубасенюк, Ред. Житомир: Вид-во ЖДУ ім. І. Франка, 2012.
140. В. Е. Радионов, "Теоретические основы педагогического проектирования", дис. доктора пед. наук, СПб, 1996.
141. С. А. Раков, В. П. Горох, "Програмно-методичний комплекс DG як крок від традиційної до інформаційної технології навчання геометрії", *Комп'ютер у школі і сім'ї*, № 1, с. 20-23, 2003.
142. Ю. С. Рамський, та М. В. Рафальська, "Формування компетентностей майбутніх вчителів інформатики та математики у галузі моделювання", *Науковий часопис НПУ імені М. П. Драгоманова. Серія 2: Комп'ютерно-орієнтовані системи навчання*, №12, с. 117-126, 2012.
143. Н. А. Рибачок, *Операційна система Linux: інструкції до виконання лабораторних робіт з дисципліни «Операційні системи». Частина 1: навч. посіб. для студ. спеціальності 121 «Інженерія програмного забезпечення», спеціалізації «Програмне забезпечення комп'ютерних та інформаційно-пошукових систем»*. К.: КПІ ім. Ігоря Сікорського, 2018.
144. "Розробка додатків та адміністрування в ОС Linux", *Офіційний сайт фізико-технічного факультету Донецького національного університету імені Василя Стуса*. [Електронний ресурс]. Доступно: <http://phys.donnu.edu.ua/uk-ua/Pages/rozrobka-dodatkov-ta-administruvannya-Linux.aspx>. Дата звернення: Січ. 31, 2019.
145. Н. И. Рыжова, "Развитие методической системы фундаментальной подготовки будущих учителей информатики в предметной области", автореф. дис. доктора пед. наук, СПб., 2000.

146. Н. И. Рыжова, Н. Ю. Королева, О. И. Ляш, "Виртуальные машины как средство обучения будущих учителей информатики сетевым технологиям", *Информатика и образование*, № 10, с. 97-99, 2007.
147. "Сайт Чернівецького Національного університету", *Офіційний сайт Чернівецького національного університету імені Юрія Федьковича*. [Електронний ресурс]. Доступно: <http://www.chnu.edu.ua/>. Дата звернення: Січ. 31, 2019.
148. О. С. Сверчевська, "Вільне програмне забезпечення у вивченні операційної системи Linux", на *VII міжнародній науково-практичній конференції "Інноваційні технології в освіті"*, Ялта, 20-22 вересня, 2010, с. 224-226.
149. О. С. Сверчевська, "Репозиторії вільно поширюваного програмного забезпечення в навчанні операційних систем і системного програмування", *Інформаційні технології і засоби навчання*, №24 (4), 2011.
150. З. С. Сейдаметова, и С. Н. Сейтвелиева, "Облачные сервисы в образовании", *Інформаційні технології в освіті*, №9, с. 105-111, 2011.
151. С. О. Семеріков, *Фундаменталізація навчання інформатичних дисциплін у вищій школі: Монографія*, К.: НПУ ім. М. П. Драгоманова, 2009.
152. Е. В. Сидоренко, *Методы математической обработки в психологи*, СПб.: ООО «Речь», 2000.
153. С. О. Сисоєва, "Педагогічні технології", в *Енциклопедія освіти*, В. Г. Кремень, Ред. К.: Юрінком Інтер, 2008, с. 661-662.
154. З. І. Слєпкань, *Методика навчання математики: Підручник*, 2-ге вид., допов. і переробл., К.: Вища шк., 2006.
155. *Современный философский словарь*, В. Е. Кемеров, Ред. М.; Бишкек; Екатеринбург, 1996.
156. А. М. Соломатин, В. И. Гам, *Теория и практика педагогического проектирования*. Омск, 2001.

157. О. М. Спірін, "Інформаційно-комунікаційні та інформатичні компетентності як компоненти системи професійно-спеціалізованих компетентностей вчителя математики", *Інформаційні технології і засоби навчання*, № 5(13), 2009.
158. О. М. Спірін, *Теоретичні та методичні засади професійної підготовки майбутніх учителів інформатики за кредитно-модульною системою: Монографія*, Житомир: Вид-во ЖДУ ім. І. Франка, 2007.
159. О. М. Спірін, "Теоретичні та методичні основи кредитно-модульної системи навчання майбутніх учителів інформатики", дис. доктора пед. наук, К., 2009.
160. О. М. Спірін, О. С. Головня, "Застосування технологій віртуалізації Unix-подібних операційних систем у підготовці бакалаврів інформатики", *Інформаційні технології і засоби навчання*, т. 65, № 3, с. 201-222, 2018.
161. О. М. Спірін, О. С. Сверчевська, "Linux та VirtualBox у навчанні абстрактних понять теорії операційних систем", на *FOSS Lviv-2011*, Львів, 1-4 лютого, 2011, с. 149-151.
162. М.Ф. Степко, Б.В. Клименко, Л.Л. ТОВАЖНЯНСЬКИЙ, *Болонський процес і навчання впродовж життя: монографія*, Х.: НТУ, 2004.
163. В. Ю. Стрельников, *Проектування професійно-орієнтованої дидактичної системи підготовки бакалаврів економіки: монографія*, Полтава: РВЦ ПУСКУ, 2006.
164. А. М. Стрюк, "Система «Агапа» як засіб навчання системного програмування бакалаврів програмної інженерії", дис. канд. пед. наук, Інститут ІТЗН НАПН України, К., 2012.
165. "Сумський державний університет", *Офіційний сайт*. [Електронний ресурс]. Доступно: <https://www.sumdu.edu.ua/ukr/>. Дата звернення: Січ. 31, 2019.
166. Н. Н. Суртаева, "Проектирование педагогических технологий в профессиональной подготовке учителя (на примере естественнонаучных дисциплин)", дис. доктора пед. наук, М., 1995.

167. *Сучасні високошвидкісні обчислювальні системи: Комп'ютерний практикум: навч. посіб. для студ. спеціальності 121 «Інженерія програмного забезпечення», уклад.: О. С. Шкурат, В. Я. Юрчишин. К.: КПІ ім. Ігоря Сікорського, 2018.*
168. Э. Таненбаум, *Современные операционные системы*, 3-е изд, СПб.: Питер, 2010.
169. Э. Таненбаум, Х. Бос, *Современные операционные системы*, 4 изд., Питер, 2015.
170. Т. М. Терещенко, К. М. Сафронов, "Використання віртуальних машин в проектуванні спеціалізованих систем", *Вісник Східноукраїнського національного університету імені Володимира Даля*, № 2, 2008.
171. Т. В. Тихонова, "Педагогічні умови професійного саморозвитку майбутнього вчителя інформатики", дис. канд. пед. наук, Інститут педагогіки АПН України, К., 2001.
172. Т. В. Тихонова, "Технологія особистісно орієнтованої освіти у професійній підготовці вчителя інформатики", *Вісник Житомирського педагогічного університету*, № 6, с. 161-163, 2000.
173. Ю. В. Триус, "Методика використання пакету Maple 7 для розв'язування екстремальних задач", *Теорія та методика навчання математики, фізики, інформатики: Збірник наукових праць*, № 5, т. 1, с. 282-296, 2005.
174. Ю. В. Триус, "Система формування інформаційної культури студентів вищих навчальних закладів як важлива складова їх професійної підготовки", *Вісник Черкаського університету, серія "Педагогічні науки"*, № 73, с. 122-130, 2005.
175. Т. І. Туркот, *Педагогіка вищої школи: навч. посібн.: реком. МОН України для студентів магістратури вищих навчальних закладів непедагогічного профілю*, Херсон, 2011.
176. О. Ю. Усата, "Підготовка майбутніх учителів інформатики до впровадження особистісно орієнтованих технологій навчання", автореф. дис. канд. пед. наук, ЖДУ ім. І. Франка, Житомир, 2009.

177. "Факультет інформаційних технологій – Київський національний університет імені Тараса Шевченка", *Офіційний сайт*. [Електронний ресурс]. Доступно: <http://fit.univ.kiev.ua/>. Дата звернення: Січ. 31, 2019.
178. "Факультет комп'ютерних наук та кібернетики", *Офіційний сайт факультету комп'ютерних наук та кібернетики Київського національного університету імені Тараса Шевченка*. [Електронний ресурс]. Доступно: <http://csc.knu.ua/uk/>. Дата звернення: Січ. 31, 2019.
179. "Факультет радіофізики, електроніки та комп'ютерних систем – Київський національний університет імені Тараса Шевченка", *Офіційний сайт*. [Електронний ресурс]. Доступно: <https://tex.knu.ua/>. Дата звернення: Січ. 31, 2019.
180. В. М. Франчук, "Віртуальні машини та їх використання", *Комп'ютер у школі та сім'ї*, № 4 (68), с. 41-44, 2008.
181. "ХНУРЕ – Харківський національний університет радіоелектроніки", *Офіційний сайт*. [Електронний ресурс]. Доступно: <http://nure.ua/>. Дата звернення: Січ. 31, 2019.
182. Г. Ю. Цибко, "Підвищення рівня теоретичної підготовки з інформатики на фізико-математичних факультетах педагогічних вузів", дис. канд. пед. наук, НПУ ім. М.П. Драгоманова, К., 1998.
183. Л. Черняк, "Средство против сложности", *Открытые системы*, № 5, 2004. [Электронный ресурс]. Доступно: <http://www.osp.ru/os/2004/05/184272/>. Дата обращения: Июнь 22, 2018.
184. "Чорноморський національний університет імені Петра Могили", *Офіційний сайт*. [Електронний ресурс]. Доступно: <https://chmnu.edu.ua/>. Дата звернення: Січ. 31, 2019.
185. В. А. Шеховцов, *Операционные системы: Учебник*. К.: Видавнича група ВНУ, 2005.
186. Г. Шилдт, *Полный справочник по C++*, 4-е изд., М.: Издательский дом "Вильямс", 2006.

187. М. П. Шишкіна, "Теоретико-методичні засади формування і розвитку хмарно орієнтованого освітньо-наукового середовища вищого навчального закладу", дис. доктора пед. наук, Інститут ІТЗН, К., 2016.
188. Г. В. Шугайло, "Розвиток композиційно-графічних умінь у майбутніх учителів інформатики у процесі диференційованого вивчення комп'ютерних технологій", *Вісник Житомирського державного педагогічного університету*, № 6, 2000, с. 199-202.
189. М. В. Шугуров, "Виртуальная герменевтика", *Труды лаборатории виртуалистики*, № 13, 2001.
190. "Як коригуються дані Трендів – Тенденції Довідка", Support.google.com. [Електронний ресурс]. Доступно: https://support.google.com/trends/answer/4365533?hl=uk&ref_topic=6248052. Дата звернення: Черв. 22, 2018.
191. Н. О. Яковлева, *Педагогическое проектирование инновационных образовательных систем: Монография*. Челябинск: Изд-во ЧГИ, 2008.
192. В. А. Ясвин, *Образовательная среда: от моделирования к проектированию: Монография*. М.: Смигал, 2001.
193. С. М. Яшанов, "Віртуальні машини в системі інформаційно-навчального середовища вищого навчального закладу", *Інформаційні технології і засоби навчання*, № 2(16), 2010.
194. "About OIT", *Princeton University Official Website, Office of Information Technology*. [Online]. Available: <http://www.princeton.edu/oit/about/>. Accessed on: June 22, 2018.
195. K. Adams, and O. Agasen, "A Comparison of Software and Hardware Techniques for x86 Virtualization", in *ASPLOS'06*, San Jose, California, USA, October 21-25, 2006.
196. "Advanced Operating Systems (L41). Course pages 2014-15". *Cambridge University Official Website*. [Online]. Available: <http://www.cl.cam.ac.uk/teaching/1415/L41/>. Accessed on: Nov. 16, 2016.
197. C. L. Anderson, M. Nguyen, "A Survey of Contemporary Instructional

- Operating Systems for use in Undergraduate Courses", *Journal of Computing Sciences in Colleges*, No. 21, pp. 183-190, 2005.
198. J. Andrus, J. Nieh, "Teaching Operating Systems Using Android", in *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education (SIGCSE'12)*, 2012.
 199. "Application Virtualization 4", *Microsoft Docs*. [Online]. Available: <https://docs.microsoft.com/en-us/microsoft-desktop-optimization-pack/appv-v4/index>
 200. A. J. Aviv, V. Mannino, T. Owlarn, S. Shannin, K. Xu, and B. T. Loo, "Experiences in Teaching an Educational User-Level Operating Systems Implementation Project", in *ACM SIGOPS Operating Systems Review*, vol. 46, no. 2, pp. 80-86, 2012.
 201. J. Bacon, and T. Harris, *Operating systems: Concurrent and distributed software design*. Addison Wesley, 2003.
 202. "Baking Pi – Operating Systems Development", *Cambridge University Official Website*. [Online]. Available: <https://www.cl.cam.ac.uk/projects/raspberrypi/tutorials/os/>. Accessed on: Nov. 16, 2016.
 203. M. Beham, M. Vlad, and H. P. Reiser, "Intrusion detection and honeypots in nested virtualization environments", in *Proceedings of the 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 24-27 June, 2013.
 204. M. Ben-Yehuda et al., "The Turtles Project: Design and Implementation of Nested Virtualization", in *Proceedings of 9th USENIX Symposium on Operating Systems Design and Implementation (OSDI 2010)*, Vancouver, BC, Canada, October 4-6, 2010, pp. 423-436.
 205. D. Brylow, "An Experimental Laboratory Environment for Teaching Embedded Operating Systems", in *Proceedings of the 39th ACM Technical Symposium on Computer Science Education (SIGCSE'08)*, New York, NY, USA, 2008, pp. 192-196.

206. E. Bugnion, S. Devine, K. Govil, and M. Rosenblum, "DISCO: Running Commodity Operating Systems on Scalable Microprocessors", *ACM Transactions on Computer Systems*, vol. 4 (15), 1997, pp. 143-156.
207. J. P. Buzen, and U. O. Gagliardi, "The evolution of virtual machine architecture", in *Proceedings of the AFIPS'73 National Computer Conference and Exposition*, New York, NY, USA, 1973, pp. 291-299.
208. "Chapter 1. Cygwin Overview". *Cygwin User's Guide*. [Online]. Available: <https://cygwin.com/cygwin-ug-net/overview.html#what-is-it>
209. W. A. Christopher, S. J. Procter, T. Anderson, "The Nachos Instructional Operating System", in *Proc. USENIX Winter 1993*, San Diego, California, USA, January 25, 1993. [Online]. Available: <http://www.eecs.berkeley.edu/Pubs/TechRpts/1993/CSD-93-739.pdf>. Accessed on: June 22, 2018.
210. Commission of the European Communities. (21.11.2001). *COM(2001) 678 final, Making a European Area of Lifelong Learning a Reality*, Brussels, 2001. [Online]. Available: <http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=COM:2001:0678:FIN:EN:PDF>. Accessed on: June 22, 2018.
211. Computing Curriculum 2005: The Overview Report. Association for Computing Machinery (ACM), Association for Information Systems (AIS), Computer Society (IEEE-CS). (2005). [Online]. Available: http://www.acm.org/education/education/curric_vols/CC2005-March06Final.pdf. Accessed on: June 22, 2018.
212. "Computer Science 318: Operating Systems (Fall 2015)", *Princeton University Official Website*. [Online]. Available: <https://www.cs.princeton.edu/courses/archive/fall15/cos318/>. Accessed on: Nov. 16, 2016.
213. Computer Science Curricula 2013. Curriculum Guidelines for Undergraduate Degree Programs in Computer Science. The Joint Task Force on Computing Curricula. Association for Computing Machinery (ACM) IEEE Computer Society. (2013). [Online]. Available: <https://www.acm.org/education/CS2013-final-report.pdf>. Accessed on: June 22, 2018.

214. "CS 140: Operating Systems (Winter 2016)", *Stanford University Official Website*. [Online]. Available: <http://web.stanford.edu/~ouster/cgi-bin/cs140-winter16/index.php>. Accessed on: Nov. 16, 2016.
215. "CS 422/522: Design and Implementation of Operating Systems", *Yale University Official Website*. [Online]. Available: <http://flint.cs.yale.edu/cs422/>. Accessed on: Nov. 16, 2016.
216. "CS 161: Operating Systems (Spring 2016)", *Harvard University Official Website*. [Online]. Available: <http://www.eecs.harvard.edu/~cs161/>. Accessed on: Nov. 16, 2016.
217. "CS162: Operating Systems and Systems Programming", *UC Berkley Official Website*. [Online]. Available: <https://cs162.eecs.berkeley.edu/>. Accessed on: Nov. 16, 2016.
218. "CS 8803-002: Introduction to Operating Systems", *Georgia Tech Official Website*. [Online]. Available: <https://www.omscs.gatech.edu/cs-8803-introduction-operating-systems>. Accessed on: Nov. 16, 2016.
219. "CSCI 432: Operating Systems (Spring 2016)", *Williams College Official Website*. [Online]. Available: <http://www.cs.williams.edu/~jeannie/cs432/index.html>. Accessed on: Nov. 16, 2016.
220. C. Dall, J. Nieh, "Teaching Operating Systems Using Code Review", in *Proceedings of SIGCSE'14*, Atlanta, GA, USA, March 5-8, 2014.
221. P. Dillenbourg, D. Schneider, and P. Synteta, "Virtual learning environments", in Proc. 3rd Hellenic Conference "Information & Communication Technologies in Education", Rhodes, Greece, 2002, pp. 3-18. [Online]. Available: <http://hal.archives-ouvertes.fr/hal-00190701/>. Accessed on: June 22, 2018.
222. S. Duignan, and T. Hall, "Using Platform Virtualisation to Teach System Architectures in Undergraduate Computer Science – An Evaluation of Student Learning Experiences", in *Innovative Techniques in Instruction Technology, E-learning, E-assessment, and Education*, M. Iskander, Ed. Dordrecht: Springer, 2008, pp. 479-484.

223. "ESXi Bare Metal Hypervisor", VMware. [Online]. Available: <https://www.vmware.com/products/esxi-and-esx.html>
224. A. Garmpis, and N. Gouvatsos, "Innovative teaching methods in Operating Systems: The Linux case", in *Innovative approaches in Education: Design and Networking*, 2012.
225. F. Giraldeau, M. R. Dagenais, and H. Boucheneb, "Teaching operating systems concepts with execution visualization", in *121st ASEE Annual Conference and Exposition*, Indianapolis, IN, USA, 2014.
226. T. Golemanov, E. Golemanova, "A Teaching in Operating Systems Tool", in *Proceedings of International Conference on Computer Systems and Technologies (CompSysTech '06)*, 2006.
227. K. Grammer, J. Stoleran, and B. Yi, "Introduction of virtualization in the teaching of operating systems for CS undergraduate program", *Journal of Computing Sciences in Colleges*, vol. 26, No. 6, pp. 44-50, 2011.
228. L. Gu, A. Vaynberg, B. Ford, Z. Shao, D. Costanzo, "CertiKOS: A Certified Kernel for Secure Cloud Computing", in *APSys'11*, Shanghai, China, July 11-12, 2011.
229. I. Habib, "Virtualization with KVM", *Linux Journal*, 2008.
230. R. Hess, P. Paulson, "Linux Kernel Projects for an Undergraduate Operating Systems Course", in *Proceedings of the 41st ACM Technical Symposium on Computer Science Education (SIGCSE'10)*, New York, NY, USA, 2010, pp. 485-489.
231. D. A. Holland, A. T. Lim, M. I. Seltzer, "A New Instructional Operating System", in *Proceedings of the 33rd SIGCSE technical symposium on Computer science education*, pp. 111-115, 2002. [Online]. Available: http://docencia.ac.upc.edu/FIB/SO/pla91/doc/OS_161.pdf. Accessed on: June 22, 2018.
232. J. Hoopes, *Virtualization for Security: Including Sandboxing, Disaster Recovery, High Availability, Forensic Analysis, and Honeypotting*, 2009.

233. "Introduction to Operating Systems (Offered at Georgia Tech as CS 8803)", *Udacity MOOC*. [Online]. Available: <https://www.udacity.com/course/introduction-to-operating-systems--ud923>. Accessed on: Nov. 16, 2016.
234. "Introduction to Operating Systems", *Udacity Wiki*. [Online]. Available: <https://www.udacity.com/wiki/ud923>. Accessed on: Nov. 16, 2016.
235. "Jails – FreeBSD Wiki", *Wiki.freebsd.org*. [Online]. Available: <https://wiki.freebsd.org/Jails>. Accessed: June 22, 2018.
236. D. Jones, A. Newman, "A Constructivist-based Tool for Operating Systems Education", in *Proceedings of EdMedia: World Conference on Educational Media and Technology*, Denver, Colorado, USA, 2002.
237. M. T. Jones, "Virtual Linux: An overview of virtualization methods, architects, and implementations", *IBM developerWorks: Technical library*. [Online] Available: <http://www.ibm.com/developerworks/library/l-linuxvirt/>. Accessed on: February 9, 2012.
238. "582219: Käyttöjärjestelmät (582219: Operating Systems)", *University of Helsinki Official Website*. [Online]. Available: <https://www.cs.helsinki.fi/en/courses/582219>. Accessed on: Nov. 16, 2016.
239. A. Kivity, Y. Kamay, D. Laor, U. Lublin, A. Liguori, "KVM: the Linux Virtual Machine Monitor" in *Proceedings of the Linux Symposium*, Ottawa, Ontario, Canada, June 27-30, 2007, vol. 1, pp. 225-230.
240. D. E. Knuth, *The Art of Computer Programming*, 3rd ed. Addison Wesley, Reading Mass, 1998.
241. "KVM", *Linux-kvm.org*. [Online]. Available: http://www.linux-kvm.org/page/Main_Page. Accessed on: June 22, 2018.
242. O. Laadan, J. Nieh, and N. Viennot, "Teaching operating systems using virtual appliances and distributed version control", in *the 41st ACM technical symposium on Computer science education (SIGCSE '10)*, 2010, pp. 480-484.
243. P. Legendre, "Coefficient of concordance", *Encyclopedia of Research Design*, N. J. Salkind, Ed., vol. 1. SAGE Publications Inc., Los Angeles, 2010., pp. 164-169.

244. Y. Li, W. Li, and C. Jiang, "A Survey of Virtual Machine System: Current Technology and Future Trends", in *Proceedings of the Third International Symposium on Electronic Commerce and Security*, 2010.
245. "Linux at Princeton", *PUG User's Guide*. [Online]. Available: http://pug.mycpanel.princeton.edu/faq/users_guide. Accessed on: June 22, 2018.
246. "Linux Containers", *Linuxcontainers.org*. [Online]. Available: <https://linuxcontainers.org/>. Accessed on: June 22, 2018.
247. J. Liu, "Deepen the Teaching Reform of Operating System, Cultivate the Comprehensive Quality of Students", *International Education Studies*, vol. 3, No. 1, pp. 151-153, 2010.
248. L. P. Maia, F. B. Machado, A. C. Pacheco Jr., "A Constructivist Framework for Operating Systems Education: a Pedagogic Proposal Using the SOsim", in *Proceedings of ITiCSE'05*, Monte de Caparica, Portugal, June 27-29, 2005.
249. J. Matlis, "Quick Study: Virtual Machines", *PC Magazine*, April 24, 2006. [Online]. Available: http://www.computerworld.com/s/article/110722/Virtual_Machines?taxonomyId=18&pageNumber=1. Accessed on: June 22, 2018.
250. P. Mell, T. Grance, "The NIST Definition of Cloud Computing", National Institute of Standards and Technology, 2011. [Online]. Available: <http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>. Accessed on: June 22, 2018.
251. N. Morze, O. Barna, O. Kuzminska, and V. Vember, "In what way should modern computer science teachers improve their teaching skills to develop students' key and IC competence?", *Open Educational E-Environment of Modern University: Collected Scientific Works*, №1, pp. 189-200, 2016.
252. S. Nanda, and T. Chiueh, "A Survey on Virtualization Technologies", *Experimental Computer Systems Lab*, SUNY Stony Brook, SUNY RPE Report TR-179, 2005.

253. J. Nieh, O. C. Leonard, "Examining VMware", *Dr.Dobb's*, August 01, 2000. [Online]. Available: <http://www.drdoobs.com/examining-vmware/184404206>. Accessed on: June 22, 2018.
254. J. Nieh, and C. Vaill, "Experiences Teaching Operating Systems Using Virtual Platforms and Linux", in *the 36th ACM Technical Symposium on Computer Science Education (SIGCSE 2005)*, 2005, pp. 520-524.
255. H. Oh, B. Kim, H. Choi, S. Moon, "Evaluation of Android Dalvik virtual machine", in *Proceedings of the 10th International Workshop on Java Technologies for Real-time and Embedded Systems (JTRES '12)*, Copenhagen, Denmark, October 24-26, 2012, pp. 115-124.
256. "OpenVZ Virtuozzo Containers Wiki", *Openvz.org*. [Online]. Available: https://openvz.org/Main_Page. Accessed: June 22, 2018.
257. "15-410, Operating System Design and Implementation", *Carnegie Mellon University Official Website*. [Online]. Available: <https://www.cs.cmu.edu/~410/>. Accessed on: Nov. 16, 2016.
258. "Operating System Engineering", *MIT OpenCourseWare*, [Online]. Available: <http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-828-operating-system-engineering-fall-2012/>. Accessed on: Nov. 16, 2016.
259. "Operating Systems I (COMS W4118)", *Columbia University Official Website*. [Online]. Available: <http://www.cs.columbia.edu/~nieh/teaching/w4118/>. Accessed on: Nov. 16, 2016.
260. "Operating Systems and Networks", *ETH Zurich Official Website*. [Online]. Available: <https://www.systems.ethz.ch/node/363>. Accessed on: Nov. 16, 2016.
261. "Operating Systems. Course pages 2014-15", *Cambridge University Official Website*. [Online]. Available: <https://www.cl.cam.ac.uk/teaching/1516/OpSystems/>. Accessed on: Nov. 16, 2016.
262. "6.828: Operating Systems Engineering", *MIT Official Website*. [Online]. Available: <https://pdos.csail.mit.edu/6.828/2014/index.html>. Accessed on: Nov. 16, 2016.

263. "Oracle VM 3: Architecture and Technical Overview", *Oracle Whitepaper*, 2016. [Online]. Available: <http://www.oracle.com/technetwork/server-storage/vm/ovm3-arch-tech-overview-459307.pdf>
264. "Oracle VM Server for x86", *oracle.com*. [Online]. Available: <https://www.oracle.com/ua/virtualization/vm-server-for-x86/index.html>. Accessed on: July 2, 2018.
265. "Oracle VM VirtualBox", *Virtualbox.org*. [Online]. Available: <https://www.virtualbox.org/>. Accessed on: June 22 2018.
266. J. Owens, "Using virtualization to teach Linux system administration in online courses", in *IBM ICVCI*, 2007.
267. A. Palicherla, T. Zhang, and D. Porter, "Teaching Virtualization by Building a Hypervisor", in *Proceedings of the 46th ACM Technical Symposium on Computer Science Education (SIGCSE 2015)*, 2015, pp. 424-429.
268. M. Pearce, S. Zeadally, and R. Hunt, "Virtualization: Issues, security threats, and solutions", *ACM Computing Surveys (CSUR)*, vol. 45, No. 2, Article 17, 2013.
269. N. Penneman et al., "Formal virtualization requirements for the ARM architecture", *Journal of Systems Architecture*, vol. 59, No. 3, pp. 144-154, 2013.
270. G. J. Popek, and R. P. Goldberg, "Formal Requirements for Virtualization Third Generation Architectures", *Communications of the ACM*, vol. 17, No. 7, pp. 412-421, 1974.
271. "Proxmox VE. Features", *Proxmox Official Website*. [Online]. Available: <https://www.proxmox.com/en/proxmox-ve>
272. "QEMU", *Wiki.qemu.org*. [Online]. Available: https://wiki.qemu.org/Main_Page. Accessed: June 22, 2018.
273. M. Rosenblum, "The Reincarnation of Virtual Machines", *Queue*, vol. 2, No. 5, pp. 34-40, 2004.
274. L. Shamir, "Teaching GUI in Operating Systems Courses", in *ACM Technical Symposium on Computer Science Education (SIGCSE 2007)*, Covington, KY,

March 2007.

275. A. Silberschatz, P. Galvin and G. Gagne, *Operating system concepts*, 9th ed., Hoboken, NJ: John Wiley & Sons, 2013.
276. A. Singh, "An Introduction to Virtualization", Kernelthread.com, 2004. [Online]. Available: <http://www.kernelthread.com/publications/virtualization/>. Accessed on: June 22, 2018.
277. J. E. Smith, and R. Nair, "The Architecture of Virtual Machines", *Computer*, vol. 38, No. 5, pp. 32-38, 2005.
278. W. Stallings, *Operating Systems Internals and Design Principles*, 8th ed., Pearson, 2015.
279. J. Sugerman, G. Venkitachalam, and B. Lim, "Virtualizing I/O Devices on VMware Workstation's Hosted Virtual Machine Monitor", in *Proceedings of the 2001 USENIX Annual Technical Conference*, Boston, Massachusetts, USA, June 25-30, 2001.
280. B. Sun, "Software Virtualization Rootkits", in Blackhat Europe 2007 Briefings & Trainings materials, Amsterdam, Netherlands, March 27-30, 2007.
281. "ThinApp Virtual Applications. Overview", *VMware Products*. [Online]. Available: <https://www.vmware.com/ru/products/thinapp.html>
282. M. Tulloch et al, *Understanding Microsoft Virtualization Solutions. From the Desktop to the Datacenter*, 2nd edition, Microsoft, 2010.
283. "Turnkey Linux". [Online]. Available: <https://www.turnkeylinux.org/>. Accessed on: June 22, 2018.
284. R. Uhlig et al., "Intel Virtualization Technology", *Computer*, vol. 38, No. 5, pp. 48-56, 2005.
285. Understanding Full Virtualization, Paravirtualization, and Hardware Assist. VMware, 2008. [Online]. Available: <https://www.vmware.com/techpapers/2007/understanding-full-virtualization-paravirtualizat-1008.html>. Accessed on: June 22, 2018.
286. "VirtualBoxes". [Online]. Available: <http://virtualboxes.org/>. Accessed on: June 22, 2018.

287. Virtualization in Education. IBM Global Education White Paper, 2007. [Online]. Available: <http://www-07.ibm.com/solutions/in/education/download/Virtualization%20in%20Education.pdf>. Accessed on: June 22, 2018.
288. "Virtualization with KVM and Linux Containers (LXC), highly available clusters, storage and more features", *Proxmox VE*. [Online]. Available: <https://www.proxmox.com/en/proxmox-ve/features>
289. "Virtuozzo 7 System Containers™, Virtual Machines, Software-Defined Storage... All in One Solution". [Online]. Available: https://www.virtuozzo.com/fileadmin/user_upload/downloads/Virtuozzo7-DS-EN-Ltr.pdf
290. VMware Player Support Center. [Online]. Available: <https://www.vmware.com/support/player.html>. Accessed on: June 22, 2018.
291. "Welcome to Linux-VServer.org", *Linux-VServer.org*. [Online]. Available: http://linux-vserver.org/Welcome_to_Linux-VServer.org. Accessed on: July 2, 2018.
292. A. Whitaker, R. Cox, M. Shaw, and S. Gribble, "Rethinking the design of virtual machine monitors", *Computer*, vol. 38, No. 5, pp. 57-62, 2005.
293. "WineHQ – Запуск Windows застосунків у Linux, BSD, Solaris та macOS". [Online]. Available: <https://www.winehq.org/>
294. "XenApp and XenDesktop – Virtual Apps and Desktops", *Citrix*. [Online]. Available: <https://www.citrix.com/products/xenapp-xendesktop/>
295. "XenServer. Open Source Server Virtualization", *xenserver.org*. [Online]. Available: <https://xenserver.org/>. Accessed on: July 2, 2018.
296. "XRDP". [Online]. Available: <http://www.xrdp.org/>
297. L. Yan, "Development and application of desktop virtualization technology", in *IEEE 3rd International Conference on Communication Software and Networks*, Xi'an, 2011, pp. 326-329.
298. X. Yuan, B. Pioro, R. Archer, Y. Li, " Teaching Operating Systems Using Visualization: A Comparative Study", in *Innovative Techniques in Instruction Technology, E-learning, E-assessment, and Education*, M. Iskander, Ed. Springer Netherlands, 2008, pp. 576-580.

ДОДАТКИ

Додаток А.

ІСТОРІЯ РОЗВИТКУ ТЕХНОЛОГІЙ ВІРТУАЛІЗАЦІЇ

Терміни «віртуалізація» та «віртуальна машина» виникли у 60-х роках ХХ ст. слідом за концепцією багатозадачності [62]. Основу для майбутнього становлення технологій віртуалізації було закладено у Массачусетському технологічному інституту (Massachusetts Institute of Technology, MIT) у процесі створення системи поділу часу Compatible Time Sharing System (CTSS, 1962) та у Манчестерському університеті (Manchester University) під час роботи над проектом Atlas (Atlas Project, 1962, один із перших у світі суперкомп'ютерів). Саме тоді було вперше використано технологію заміщення сторінок на вимогу та технологію викликів супервізора [237].

Т. М. Джонс [237] виділяє три основні історичні напрями розвитку технологій віртуалізації: віртуалізація обладнання, віртуалізація процесора та командна віртуалізація.

Напрямок віртуалізації обладнання. Вперше віртуалізація була реалізована на комп'ютерах IBM System 360 [62], а невдовзі – на IBM System 370 [168]. Запуск віртуальних машин та керування ними здійснювала програма CP 67, котра, у свою чергу, була частиною системи поділу часу CP/CMS. CP/CMS (пізніше перейменована на VM/370) розроблялася групою з наукового центру IBM Scientific Center, що в Кембриджі, штат Массачусетс [168].

Розглянемо механізм віртуалізації у VM/370. У VM/370 кожна віртуальна машина являла собою образ усього апаратного забезпечення фізичної машини. Кожний користувач такої системи працював з одним із таких образів, причому багато користувачів могли взаємодіяти кожний зі своїм образом паралельно, *одночасно*, що й лежало в основі реалізації режиму *поділу часу* – технології, котра поклала початок концепції багатозадачності [168].

У [168] виділено два рівні системи поділу часу, кожен з яких реалізує окрему функцію. Рівень 1 реалізує багатозадачність, а рівень 2 надає користувачам розширену машину зі зручним інтерфейсом.

На рівні 1 міститься монітор віртуальних машин (програма CP 67). Монітор віртуальних машин запускається безпосередньо на обладнанні і замість однієї реальної фізичної машини надає верхньому рівню декілька віртуальних машин, котрі є точними копіями реальної машини.

На рівні 2 на кожній з віртуальних машин працюють операційні системи. У System 360 це могла бути лише система пакетної обробки. Натомість у System 370 операційні системи не обов'язково мусили бути однаковими. Скажімо, одні користувачі могли запускати операційну систему пакетної обробки, а інші – однокористувацьку інтерактивну систему.

Нехай на рівні 2 працює однокористувацька інтерактивна система CMS, і нехай програма під управлінням CMS виконує системний виклик. Тоді спочатку виклик перехоплюється у системне переривання самою системою CMS. Далі система CMS видає низькорівневі команди до свого віртуального обладнання (наприклад, для читання свого віртуального диску). Система VM/370 перехоплює ці команди і виконує їх уже на реальному обладнанні, але так, що для CMS створюється ілюзія, ніби насправді команди виконуються на її віртуальному обладнанні [168].

Напрямок віртуалізації процесора. Інше застосування віртуальних машин – виконання псевдокоду (Р-коду). Програма, написана на псевдокодi, могла працювати тільки на спеціальній віртуальній машині, призначеній для виконання псевдокоду, натомість така програма працювала у будь-якій системі за єдиної умови – наявності доступу до віртуальної машини для виконання Р-коду [237]. Ідея Р-коду була реалізована зокрема у таких мовах програмування, як USCD Pascal (USCD є аббревіатурою від англ. University of California, San Diego – Каліфорнійський університет, Сан Дієго) та BCPL (скор. від англ. Basic Combined Programming Language – базова комбінована мова програмування, розробник Мартін Річардс, Кембриджський університет [186,

с. 30]). Код спершу переводився з BCPL у спеціальний допоміжний код для віртуальної машини, а з допоміжного коду – у машинну мову цільової машини. Саме таку модель використовують сучасні компілятори для забезпечення сумісності. Сучасним прикладом віртуалізації процесора є віртуальна машина Java (JVM – Java Virtual Machine) [237].

Напрямок командної віртуалізації. Командна, або бінарна, віртуалізація передбачає переведення (трансляцію) віртуальних команд у фізичні команди основного обладнання, зазвичай динамічним чином. Така технологія реалізована, зокрема, у процесорі Crusoe виробництва Transmeta; використовується під час сканування коду у середовищі виконання (runtime environment) з метою виявлення і перескеровування привілейованих команд (щоб працювати в конкретному середовищі конкретного процесора) [237].

Повернемося до першого історичного напрямку віртуалізації – віртуалізації обладнання. Нагадаємо, що CP/CMS належала до систем поділу часу і потребувала застосування технології віртуалізації саме тому, що вимагала створення для кожного користувача окремої віртуальної копії обладнання. Для запуску віртуального програмного забезпечення передбачалося, що процесор здатний забезпечити підтримку такої технології, оскільки деякі інструкції (т.з. привілейовані інструкції) можуть виконуватися лише на найвищому рівні привілеїв [270].

Така проблема не поставала у випадку IBM System 360, де монітор віртуальних машин працював на тому самому рівні привілеїв, що й операційна система. Тому монітор віртуальних машин міг здійснювати перехоплення привілейованих інструкцій [62].

Натомість на комп'ютерах IBM System 370 операційна система працювала на рівень вище за монітор віртуальних машин, маючи при цьому менше привілеїв. В IBM System 370 проблема вирішувалася на апаратному рівні: перехоплення привілейованих інструкцій та виклик монітора віртуальних машин для здійснення програмної емуляції даної інструкції здійснював центральний процесор [168].

Проте із відходом у минуле систем поділу часу перестала бути очевидною потреба в апаратній підтримці віртуалізації, і згодом таку підтримку не було реалізовано у процесорах x86. Пізніше це зробило віртуалізацію x86-систем дуже проблематичною.

Розглянемо суть проблеми віртуалізації x86-систем детальніше. Операційні системи, створені для процесорів x86, розраховані на роботу власне на апаратному забезпеченні. Крім того, в основі функціонування такої операційної системи лежить припущення, що система одноосібно розпоряджається всім апаратним забезпеченням комп'ютера.

В архітектурі x86 існує чотири рівні повноважень (рис. А.1), відомі також як кільця 0, 1, 2, 3. Інструкція, що виконується у кільці 0, має максимальні привілеї. У кільці 0 працює ядро операційної системи, а тому відповідний режим роботи процесора називається режимом ядра (англ. kernel mode), або привілейованим режимом (англ. privileged mode).

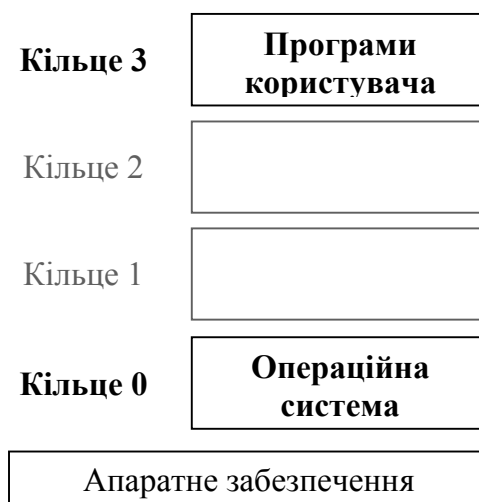


Рис. А.1. Рівні повноважень архітектури x86 без віртуалізації [285, с. 3]

Кільця 1 і 2 зарезервовані й наразі не використовуються, тимчасом як інструкції, що виконуються у кільці 3, наділені мінімальним рівнем повноважень. У кільці 3 функціонують програми користувача, тож відповідний режим роботи процесора називається режимом користувача [285].

Якщо систему на базі архітектури x86 віртуалізують, то передбачається наявність додаткового прошарку між операційною системою та апаратним

забезпеченням – віртуалізаційного прошарку. Згідно з [285], віртуалізаційний прошарок покликаний забезпечувати створення віртуальних машин та подальше керування ними, зокрема розподіл між віртуальними машинами апаратних ресурсів реальної машини.

Проте частина команд центрального процесора, необхідна для роботи операційної системи, розрахована на виконання лише у кільці 0, і ці команди не будуть виконані, якщо надійдуть від процесу, запущеного у кільці 3 [285]. Такі команди надалі називатимемо службовими командами (англ. sensitive instructions). Виділяють також привілейовані команди (англ. privileged instructions) – команди, котрі при виконанні у режимі користувача призводять до перехоплення управління [270].

Дж. Попек і Р. Голдберг у своїй роботі [270] показали, що машина може бути віртуалізована лише тоді, коли службові команди є підмножиною привілейованих команд. Е. Таненбаум [168, с. 663] спрощено трактує це твердження наступним чином: віртуалізація можлива, коли будь-яка дія у режимі користувача, котра не повинна виконуватися у цьому режимі, призводить до перехоплення управління апаратурою. На комп'ютерах з архітектурою x86 таке перехоплення не відбувається. Отож тривалий час для даної архітектури не існувало технології реалізації віртуальних машин, окрім інтерпретаторів для Pentium, недоліком яких є значні (у 5-10 разів) втрати продуктивності [168, с. 98-99].

Ситуація змінилася у 90-х роках XX ст. завдяки ряду проектів, а надто проекту Disco (Стенфордський університет) [201; 168; 206].

Велику роль у становленні віртуалізації архітектури x86 відіграли розробки компанії VMware. Свою технологію у VMware назвали повною віртуалізацією, реалізованою «поеднанням прямого виконання та бінарної трансляції» [285, с. 6]. Згідно з даною технологією, гіпервізор працює у кільці 0, що забезпечує ізоляцію віртуальних машин одна від одної та продуктивність. Віртуалізована операційна система переміщується у кільце 3, де, однак, має більше повноважень за звичайну програму, запущену у кільці 3, але менше

повноважень, ніж гіпервізор у кільці 0. Це поклало початок програмній віртуалізації.

Апаратну підтримку віртуалізації на рівні процесора було відроджено у 2004-2005 рр. компаніями Intel та AMD [4]. У процесорах Intel Core 2 технологія віртуалізації має назву Intel VT (VT – скорочено від англ. Virtual Technology – технологія віртуалізації), а у процесорах AMD Pacifica – AMD SVM (SVM – скорочено від англ. Secure Virtual Machine – безпечна віртуальна машина) [168] або просто AMD-V [62]. Спільним моментом цих двох загалом досить відмінних між собою технологій є створення контейнерів для віртуальних машин на апаратному рівні [168].

Важливо відзначити, що у період тимчасового відходу апаратної підтримки віртуалізації на рівні процесора у зв'язку з особливостями архітектури x86, компанія IBM продовжувала реалізовувати підтримку віртуалізації (System/390, z-серії). Втім, йдеться не про комп'ютери для широкого вжитку, а про великі обчислювальні потужності (наприклад, мейнфрейми) [169, с. 530].

Дослідження швидкодії різноманітних програмних та апаратних реалізацій віртуальних машин [195] показали, що апаратна реалізація не завжди швидкодійніша за програмну. Тому з метою досягнення оптимального результату доцільно за наявності відповідних апаратних можливостей поєднувати ці дві технології [168], що й відбувається на практиці у сучасних засобах віртуалізації.

Додаток Б.

АНАЛІЗ ПІДХОДІВ ДО СИСТЕМАТИЗАЦІЇ ТЕХНОЛОГІЙ ВІРТУАЛІЗАЦІЇ

Підхід Е. Таненбаума. Е. Таненбаум виділяє віртуалізацію на основі гіпервізорів I типу та гіпервізорів II типу, а також окремо розглядає паравіртуалізацію [168].

Гіпервізор I типу працює власне на апаратному забезпеченні (рис. Б.1 а) і є єдиною програмою, що працює у режимі ядра. Гіпервізор створює і підтримує декілька образів апаратного забезпечення, тобто декілька віртуальних машин. На кожную віртуальну машину встановлюється своя ОС, що називається **гостьовою ОС**. Гостьові ОС на різних віртуальних машинах не обов'язково повинні співпадати. Тим часом, **гіпервізор II типу** є користувацькою програмою, запущеною під управлінням ОС (рис. Б.1 б). ОС, у межах якої працює гіпервізор, називається **основною ОС**, а ОС віртуальних машин, як і у попередньому випадку, – гостьовими ОС.

Гіпервізори I і II типів, відзначає Е. Таненбаум, працюють із

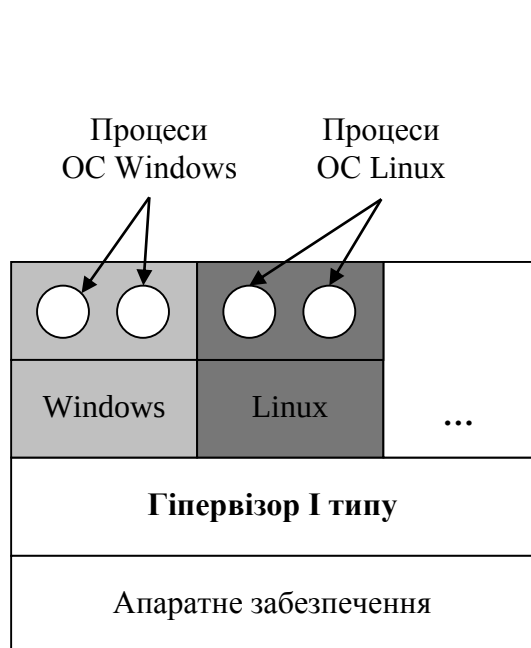


Рис. Б.1 а. Гіпервізор I типу
[168, с. 99]

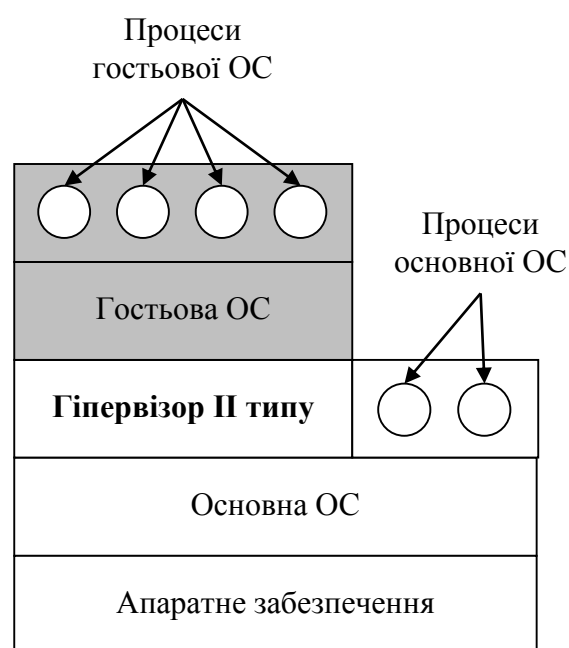


Рис. Б.1 б. Гіпервізор II типу
[168, с. 99]

не модифікованою гостьовою ОС, тобто до коду ОС не вноситься жодних змін. Натомість технологія **паравіртуалізації**, навпаки, передбачає певну модифікацію коду гостьової ОС. У коді такої ОС виклики службових команд замінюються на виклики гіпервізора. Гіпервізор, у свою чергу, надає інтерфейс, що забезпечує перехоплення та обробку таких викликів. Ця технологія є дуже близькою до інтерфейсу прикладного програмування API.

Проте паравіртуалізація може бути здійснена лише для тих ОС, програмні коди яких є відкритими. Тому паравіртуалізація, наприклад, доступна для вільно поширюваних дистрибутивів Linux і недоступна для Windows.

Таким чином, систематизацію технологій віртуалізації, описаних Е. Таненбаумом, можна схематично подати так, як це зроблено на рис. Б.2.

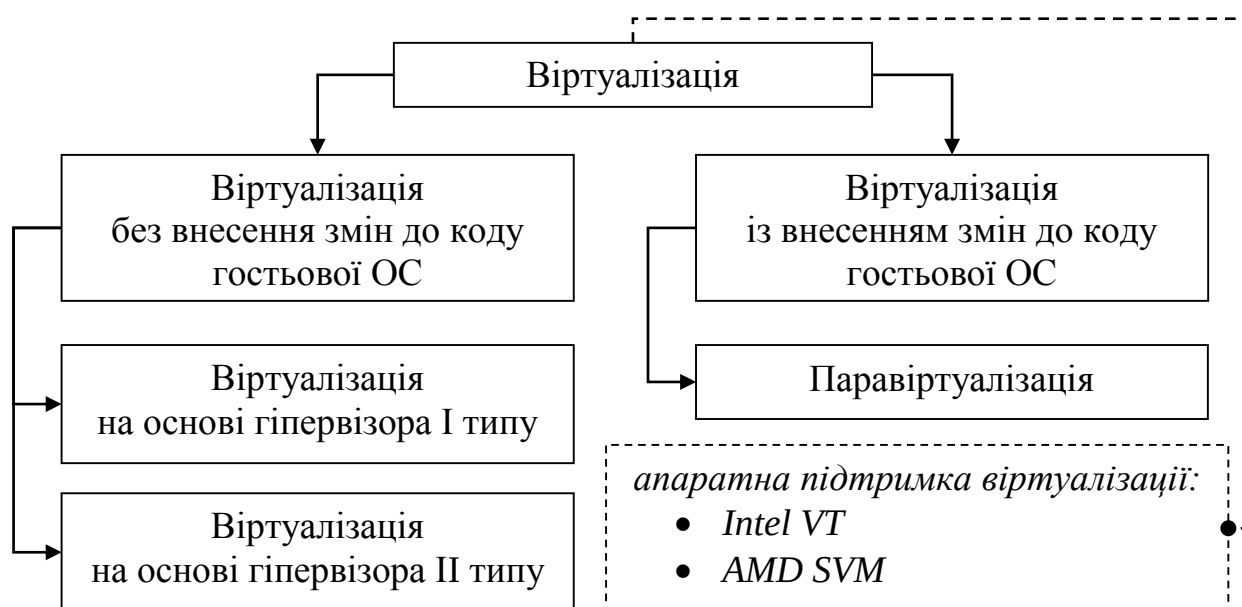


Рис. Б.2. Систематизація технологій віртуалізації: складено за підходом Е. Таненбаума

Підхід О. Батиря. У статті [4] виділено *технологічні підходи до віртуалізації* (віртуалізація серверів, віртуалізація настільних ОС, віртуалізація програмних застосунків, віртуалізація подань), причому в межах деяких із них також виділяють окремі підходи. Схематично дану систематизацію можна подати так, як це зроблено на рис. Б.3.

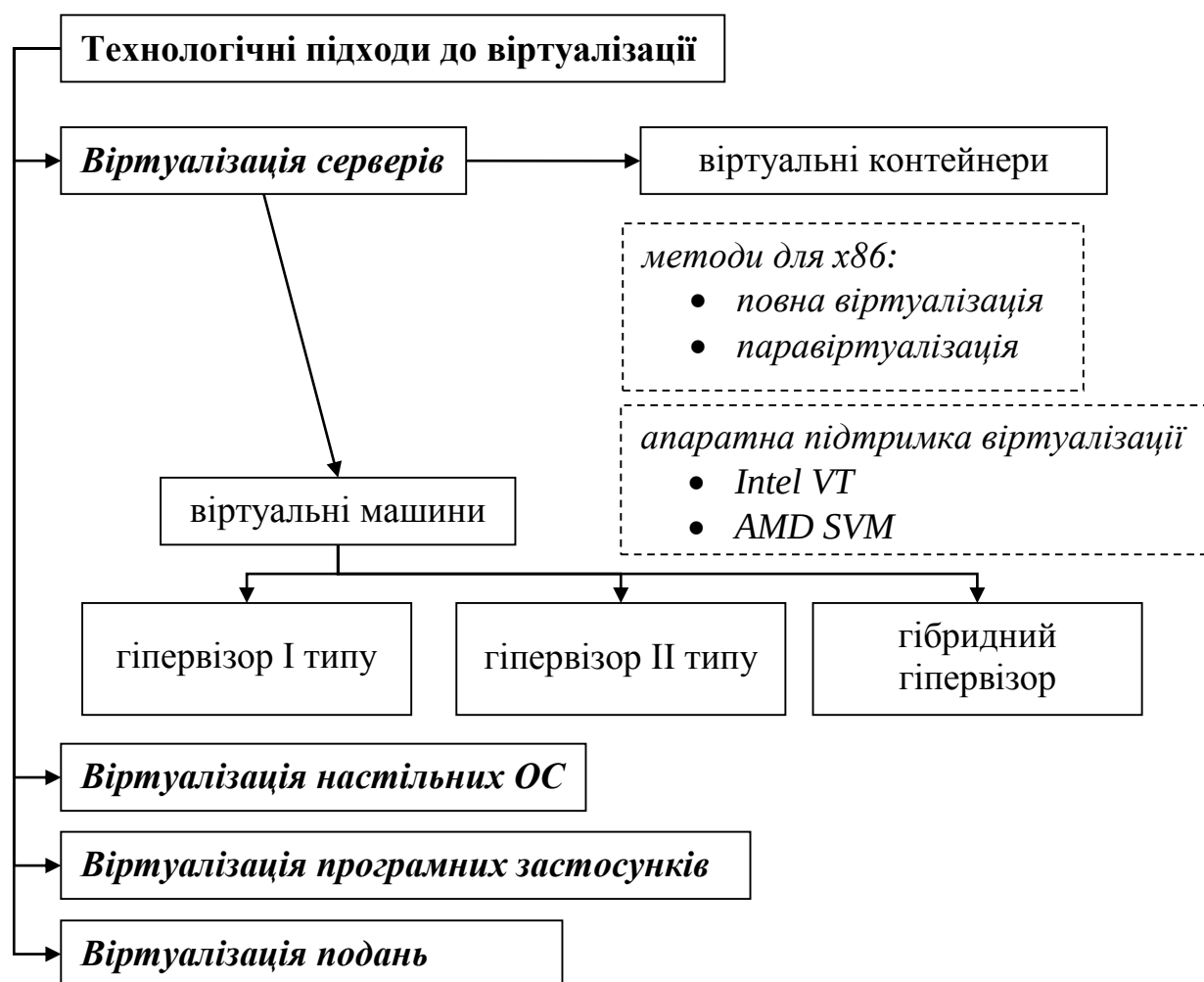


Рис. Б.3. Систематизація технологій віртуалізації:
складено за підходом О. Батира

Віртуалізація серверів (англ. server virtualization) полягає у забезпеченні можливості запуску на одному комп'ютері декількох операційних середовищ. Два основних підходи до реалізації такої можливості – це віртуальні машини та віртуальні контейнери.

Підхід віртуальних машин (англ. virtual machines) передбачає, що кожне операційне середовище (кожна віртуальна машина) працює з окремим ядром ОС, тимчасом як згідно з *підходом віртуальних контейнерів* (англ. virtual containers), усі операційні середовища (усі контейнери) працюють з ядром однієї ОС, але у власному оточенні. Зрозуміло, що у першому випадку, як уже зазначалося раніше, операційні системи різних віртуальних машин можуть бути

різними, натомість у другому випадку всі контейнери використовують ядро однієї операційної системи.

У [4], як і у [168], засіб для виконання віртуальних машин називають гіпервізором. Також тут виділено три різновиди гіпервізорів: гіпервізор I типу, гіпервізор II типу та гібридний гіпервізор.

Опис гіпервізорів I типу (автономний гіпервізор, тонкий гіпервізор, «виконуваний на голому залізі» гіпервізор, англ. native virtualization, bare-metal virtualization, Type 1, рос. нативная виртуализация) загалом відповідає аналогічному опису гіпервізорів I типу у Е. Таненбаума. Автори статті також відзначають, що такий гіпервізор фактично є мінімальною операційною системою. Як приклад гіпервізора I типу наведемо VMware ESX. Схематичне зображення гіпервізора I типу подано на рис. Б.4 а.

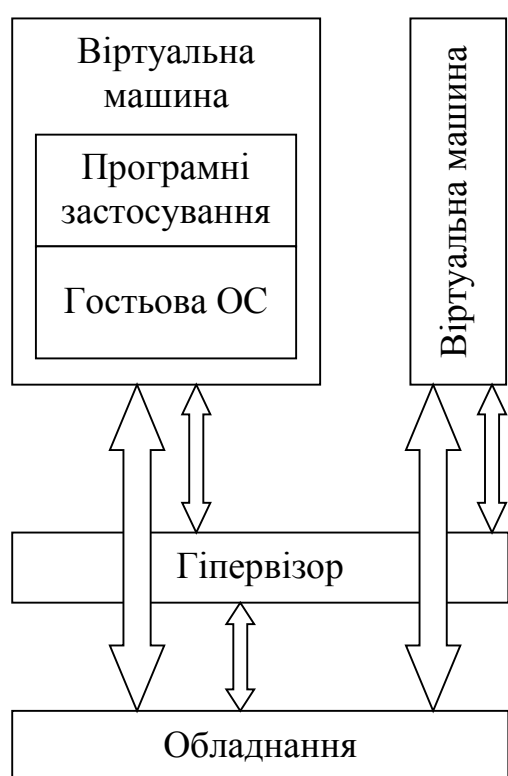


Рис. Б.4 а. Гіпервізор I типу [4]



Рис. Б.4 б. Гіпервізор II типу [4]

Гіпервізори II типу (хостовий гіпервізор, монітор віртуальних машин, англ. hosted hypervisor, virtual machine monitor, Type 2) означаються як

«спеціальний додатковий програмний прошарок, розташований поверх основної ОС» (рис. Б.4 б). До гіпервізорів II типу автори відносять Oracle VirtualBox, VMware Workstation.

Гібридні гіпервізори (англ. *hybrid hypervisor, Type 1+*) є поєднанням гіпервізорів I і II типу. Дана технологія передбачає, з одного боку, наявність тонкого гіпервізора, що працює безпосередньо на апаратному забезпеченні (як у гіпервізорах I типу), а з іншого боку – спеціальної депривілейованої сервісної операційної системи, що виконує частину функцій основної операційної системи, однак, на відміну від основної операційної системи, працює під управлінням тонкого гіпервізора. Найчастіше тонкий гіпервізор, окрім керування сервісною ОС, виконує роль засобу управління процесором і пам'яттю, тимчасом як решта функцій відводиться сервісній ОС. У ролі прикладів гібридних гіпервізорів у [4] наводять Microsoft Hyper-V, Citrix Xen Server, Oracle VM, Sun Logical Domains Hypervisor.

Також у [4] виділено два основних *методи віртуалізації* x86-систем: повна віртуалізація (бінарна трансляція, англ. *full virtualization, native virtualization*) та паравіртуалізація (англ. *paravirtualization*).

Підхід віртуальних контейнерів, або віртуалізація рівня операційної системи (англ. *operating system level virtualization*), передбачає віртуалізацію на рівні ядра операційної системи з метою створення кількох незалежних віртуальних операційних середовищ (віртуальних контейнерів), що працювали б паралельно. Як уже зазначалося вище, у такому разі усі контейнери використовують одну операційну систему, до того ж спільно використовують єдине ядро. Це зокрема означає, що неможливо мати в одному контейнері одну ОС, а в іншому – іншу (на відміну від підходу віртуальних машин). Проте, завдяки роботі з єдиним ядром, контейнерам притаманна вища швидкодія, ніж віртуальним машинам. Прикладами реалізації підходу віртуальних контейнерів є Parallels Virtuozzo, Sun Solaris Containers, Linux-VServer, OpenVZ.

Віртуалізація настільних ОС може здійснюватися з метою розробки програмного забезпечення, дослідження програмного забезпечення,

демонстрації програмного забезпечення тощо. На відміну від віртуалізації серверів, у віртуалізації настільних ОС основною задачею є не забезпечення максимальної ефективності обчислювальних ресурсів, а підтримка дуже широкого спектру апаратних пристроїв [4]. Для віртуалізації настільних ОС застосовується підхід віртуальних машин (гіпервізор II типу). Зазначимо, що термін «віртуалізація настільних ОС» може бути перекладено як «desktop virtualization», однак вважаємо за краще уникати такого перекладу, оскільки відповідний англomовний термін зазвичай застосовують у зв'язку з *віртуалізацією подань*, про яку йтиметься далі.

Метою **віртуалізації програмних застосунків** (англ. application virtualization) є забезпечення надійної роботи у багатозадачному середовищі.

Віртуалізація програмних застосунків передбачає реалізацію віртуального середовища для кожного екземпляра програми. У межах такого віртуального середовища окремо для екземпляра програми створюються файли, системний реєстр, шрифти, файли налаштувань, фонові процеси тощо. У [4] проведено паралель між віртуалізацією програмних застосунків та віртуальними контейнерами. В обох випадках маємо справу зі спільним ядром операційної системи, проте у випадку віртуальних контейнерів віртуалізація здійснюється для всіх програм, запущених у межах конкретного контейнера, а для програмних застосувань віртуалізація здійснюється для окремого екземпляра програми. Віртуалізовані описаним вище чином програмні застосунки не потребують встановлення. Віртуалізація програмних застосунків забезпечує спрощення розгортання настільних систем та подальше керування ними, а також мінімізує можливі конфлікти між програмними засобами.

Приклади реалізації віртуалізації програмних застосунків: Microsoft Application Virtualization (раніше Softgrid), VMware ThinInstall, Symantec/Altiris Virtualization, Novell ZENworks Application Virtualization.

Віртуалізація подань (англ. presentation virtualization) спрямована на забезпечення термінального доступу. У такому разі віртуалізується інтерфейс користувача: користувач взаємодіє з системою зі свого терміналу за допомогою

віртуального інтерфейсу, тимчасом як насправді процеси користувача виконуються на серверному комп'ютері. Віртуалізація подань застосовуються для залучення застарілої техніки та успадкованого програмного забезпечення, для організації безпеки та здешевлення адміністрування системи тощо.

Приклади віртуалізації подань: XenApp (раніше Presentation Server, компанія Citrix), Windows Terminal Services (Microsoft), vSpace (NComputing).

Підхід Н. З. Єлманової та С. О. Пахомова. Н. З. Єлманова та С. О. Пахомов [62] виділяють програмну та апаратну віртуалізацію (рис. Б.5). До програмної вони відносять віртуалізацію за методом бінарної (динамічної) трансляції та паравіртуалізацію (віртуалізацію за методом модифікації гостьової ОС). Апаратну технологію у даній систематизації представляють вже згадувані раніше технології Intel VT та AMD-V.

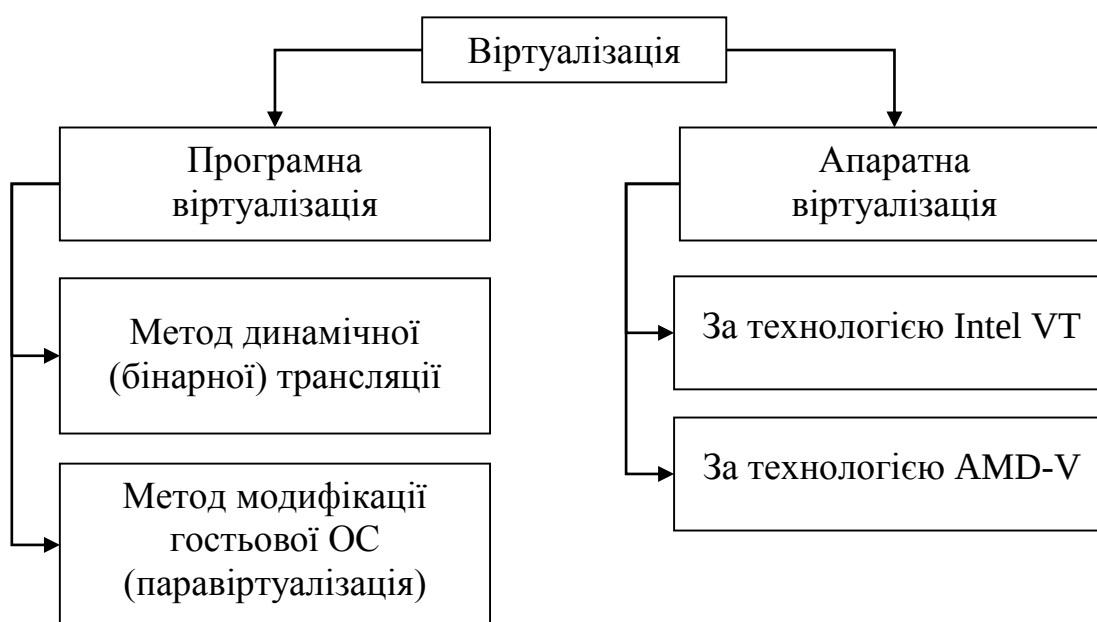


Рис. Б.5. Систематизація технологій віртуалізації: складено за підходом Н. З. Єлманової та С. О. Пахомова

Підхід Я. Метліса. Я. Метліс [249] виділяє віртуальні машини-гіпервізори (англ. *hypervisor VMs*, відповідник гіпервізорів I типу), хостові віртуальні машини (англ. *hosted VMs*, відповідник гіпервізорів II типу), віртуальні машини рівня програмних застосунків (англ. *application-level VMs*). Окремо розглянуто **паралельні віртуальні машини** (англ. *parallel VMs*). Мета паралельної віртуальної машини полягає не у створенні кількох віртуальних

комп'ютерів на основі одного фізичного, а навпаки, в об'єднанні багатьох фізичних комп'ютерів, підключених до мережі, в один віртуальний комп'ютер з багатьма процесорами. Прикладом реалізації даної технології є система PVM (Parallel Virtual Machine).

Схематичне подання систематизації за Метлісом наведено на рис. Б.6.

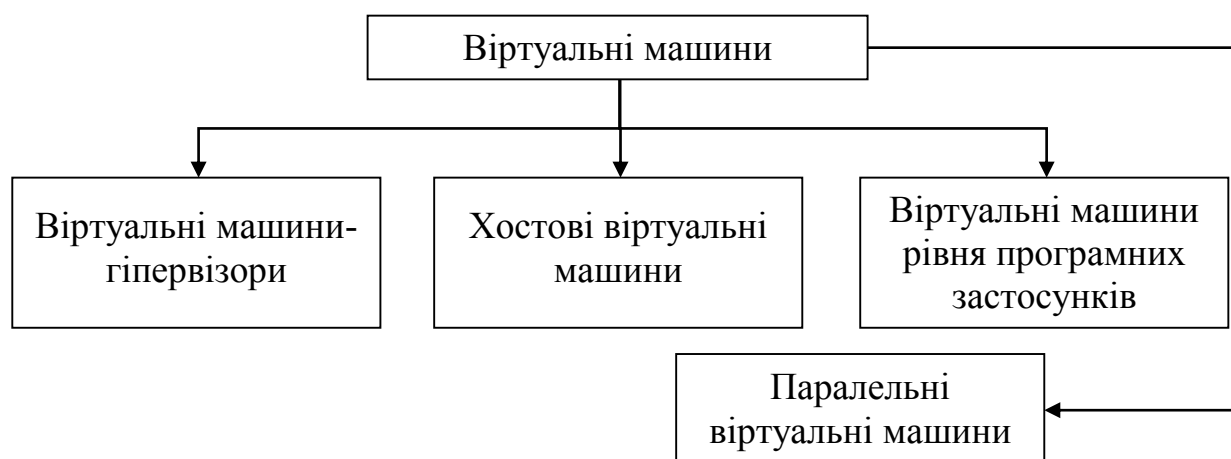


Рис. Б.6. Систематизація технологій віртуалізації: складено за підходом Я. Метліса

Підхід Б. Сана. Сан Бінг [280] виокремлює такі *технології віртуального виконання*, як повний, або строгий, емулятор (англ. pure emulator), API-емулятор (англ. API emulator) та віртуальні машини, а також окремо згадує про апаратну віртуалізацію (англ. hardware based virtualization). Схематичний вигляд даної систематизації подано на рис. Б.7.

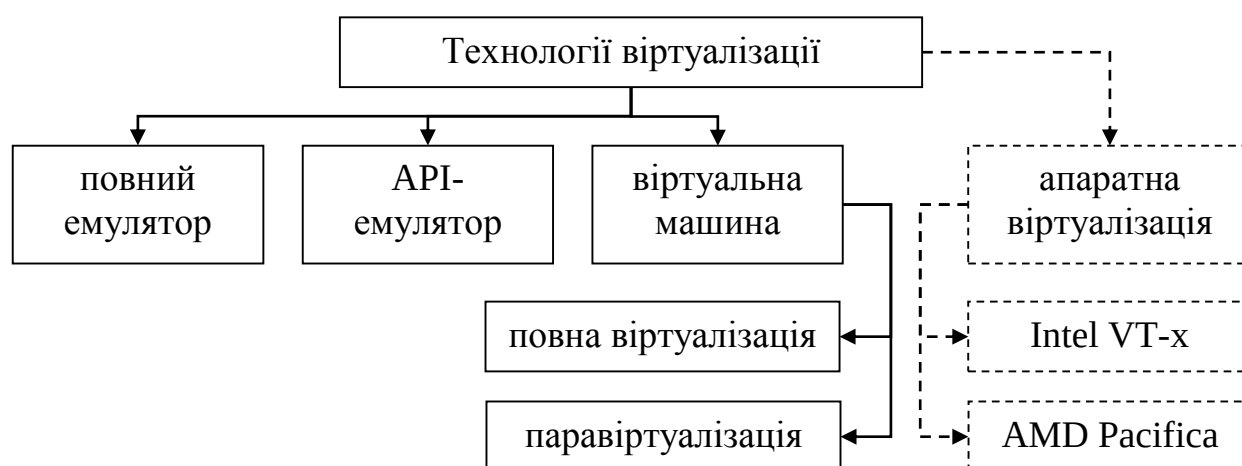


Рис. Б.7. Систематизація технологій віртуалізації: складено за підходом Б. Сана

Створення **повного емулятора** передбачає програмну емуляцію всього апаратного забезпечення, включаючи процесор. Перевагою такого підходу є можливість емуляції комп'ютера однієї архітектури на комп'ютері іншої архітектури, а недоліком – невисока продуктивність. У ролі прикладу повного емулятора наводиться програмний засіб Vochs.

API-емуляція покликана дозволити програмному коду, скомпільованому для однієї операційної системи, працювати в іншій операційній системі. Для досягнення описаної вище мети API-емулятор перехоплює API-виклики програми та емулює їх за допомогою API-викликів поточної операційної системи. API-емулятор має вищу продуктивність, ніж повний емулятор, однак його розробка для конкретної операційної системи має враховувати велику кількість деталей реалізації цієї системи, а тому API-емулятори мають низьку портативність. Прикладом втілення даної технології є програмний засіб Wine.

Віртуальні машини, зазначає Сан Бінг, покликані забезпечити можливість одночасного запуску кількох операційних систем на одному комп'ютері. При цьому принцип роботи віртуальних машин подібний до принципу роботи повних емуляторів: для реалізації всього віртуального апаратного забезпечення, окрім процесора, застосовується технологія повної емуляції. Натомість для віртуалізації процесора використовується пряме виконання у поєднанні з емуляцією, що є проявом квазіемуляції (англ. quasi-emulation technology). Технологія квазіемуляції тут полягає у наступному: більшість інструкцій виконуються звичайним чином, і лише невелика підмножина службових інструкцій перехоплюються та емулюються. У ролі прикладу технології віртуальних машин Б. Сан наводить продукти VMware.

Підхід О. К. Гультьєва. О. К. Гультьєв [54] виділяє три *схеми віртуалізації*: емуляція API гостьової ОС, повна емуляція гостьової ОС та квазіемуляція гостьової ОС (рис. Б.8). Ця систематизація дещо вужча за попередню, натомість проілюстрована більшою кількістю прикладів. Це, зокрема, Simics, Qemu (повна емуляція гостьової ОС); DOSEMU, UML,

Virtuozzo (емуляція API гостьової ОС); VMware, Serenity Virtual Station, Plex86, L4Ka, Xen (квазіемуляція гостьової ОС).

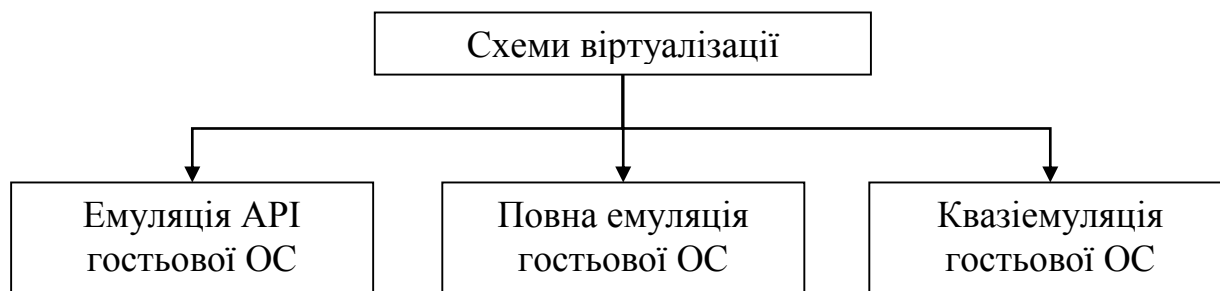


Рис. Б.8. Систематизація технологій віртуалізації: складено за підходом О. К. Гульцяєва

Підхід М. Т. Джонса. М. Т. Джонс [237] розрізняє такі *типи віртуалізації*, як емуляція обладнання (аналог повної емуляції), повна віртуалізація (аналог гіпервізорів усіх типів), паравіртуалізація та віртуалізація рівня ОС (рис. Б.9).

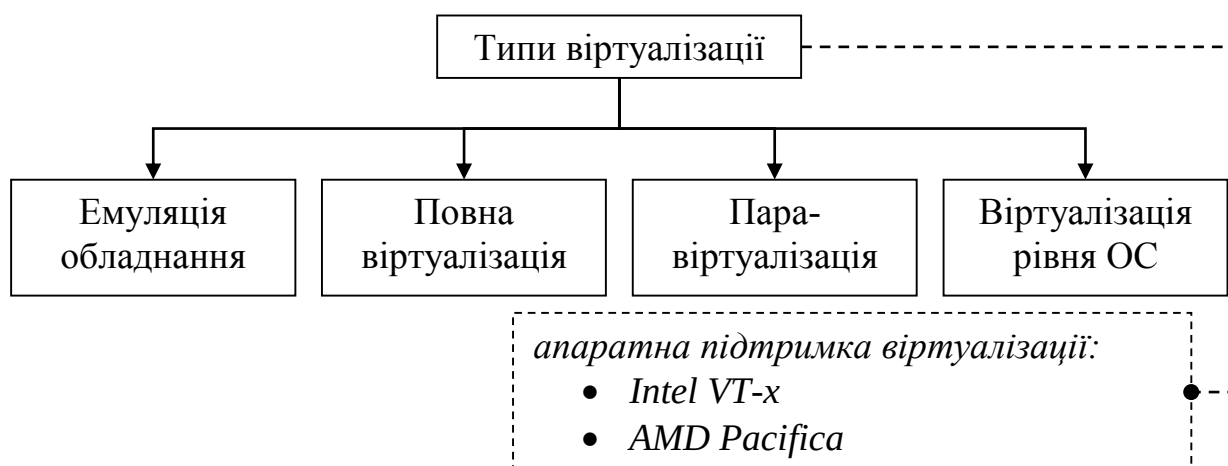


Рис. Б.9. Систематизація технологій віртуалізації: підхід М. Т. Джонса (адаптована авторська схема)

Підхід М. Розенблума (Mendel Rosenblum). М. Розенблум [273] виокремлює віртуалізацію рівня апаратного забезпечення (hardware-level virtualization), віртуалізацію рівня ОС (operating system-level virtualization) та віртуальні машини мов програмування високого рівня (high-level language virtual machines). Як приклади віртуалізації рівня апаратного забезпечення автор вказує VM370 і програмні продукти компанії VMware, як приклади віртуалізації рівня ОС – контейнери FreeBSD jail, як приклади віртуальних

машин мов програмування високого рівня – віртуальні машини Smalltalk і Java (рис. Б.10).



Рис. Б.10. Систематизація технологій віртуалізації: складено за підходом М. Розенблума

Підхід авторів С. Нанда (Susanta Nanda), З. Чуйе (Tzi-cker Chiueh) [252]

передбачає поділ віртуалізації на віртуалізацію рівня інструкцій архітектури (virtualization at the instruction set architecture level), віртуалізацію рівня обладнання (virtualization at the hardware abstraction level, або hardware virtualization), віртуалізацію рівня ОС (virtualization at the OS level), віртуалізацію рівня мови програмування (virtualization at the programming language level), віртуалізацію рівня системних бібліотек (virtualization at the library level) та полегшені віртуальні машини (lightweight virtual machines) – рис. Б.11.

До прикладів віртуалізації рівня інструкцій архітектури автори відносять Bochs, Crusoe, QEMU, BIRD, Plex86, до прикладів віртуалізації рівня ОС – FreeBSD jail, Linux Kernel-Mode Virtualization, Ensim VPS, до прикладів віртуалізації рівня мови програмування – Java Virtual Machine, Microsoft .NET CLI, Parrot, до прикладів віртуалізації рівня системних бібліотек – WINE, WABI, LxRun, Visual Main Win, до прикладів полегшених віртуальних машин – Featherweight Virtual Machine. Віртуалізацію рівня обладнання запропоновано поділити на двійкову трансляцію (binary translation), паравіртуалізацію (paravirtualization) та віртуальні машини у просторі користувача (user-space virtual machines).

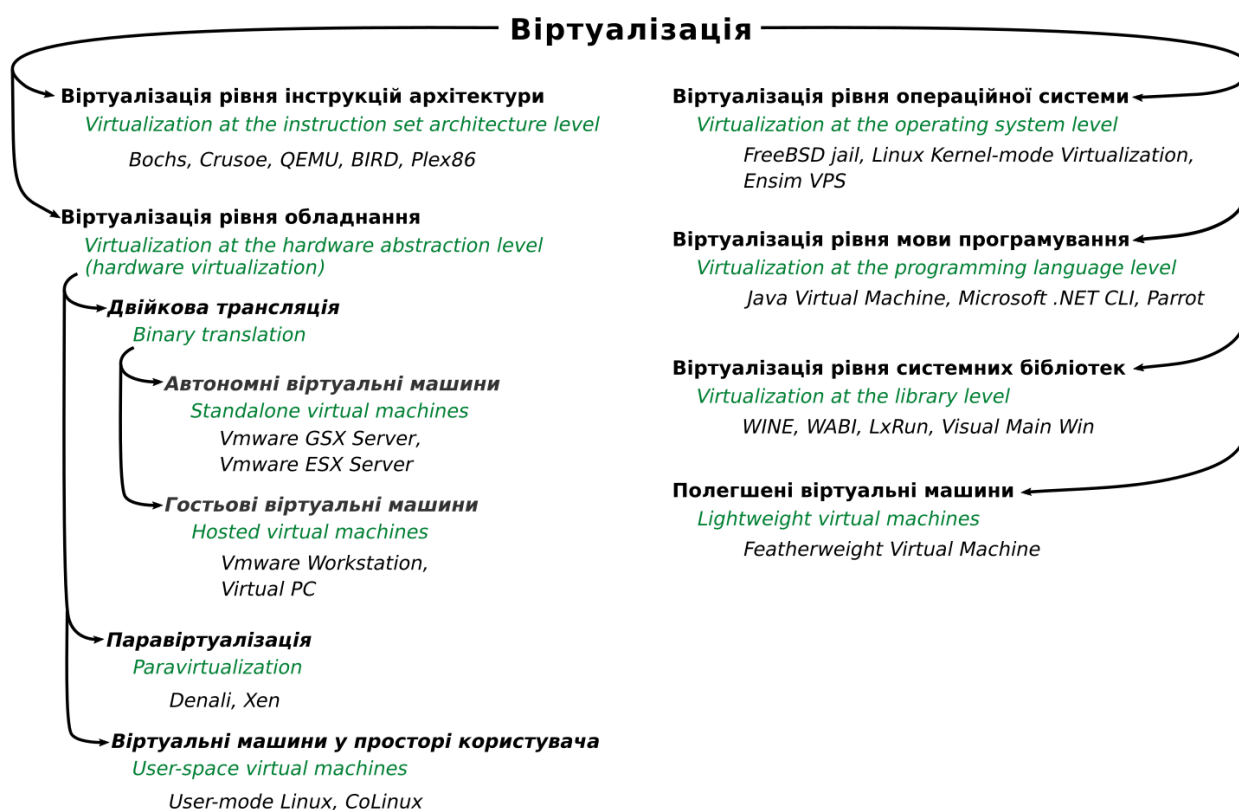


Рис. Б.11. Систематизація технологій віртуалізації:
складено за підходом авторів С. Нанда та З. Чіуе

Серед технологій двійкової трансляції автори виокремлюють автономні віртуальні машини (standalone virtual machines), наприклад, Vmware GSX Server, Vmware ESX Server, а також гостьові віртуальні машини (hosted virtual machines), приміром, Vmware Workstation, Virtual PC.

Наведено такі приклади технологій паравіртуалізації, як Denali, Xen, і такі приклади віртуальних машин у просторі користувача, як User-mode Linux, CoLinux.

Підхід Дж. Сміта (James E. Smith) та Р. Наїра (Ravi Nair). Дж. Сміт та Р. Наїр [277] поділяють віртуальні машини на процесові віртуальні машини (process virtual machines) і системні віртуальні машини (system virtual machines) – див. рис. Б.12. У межах обох згаданих вище різновидів віртуальних машин виокремлено віртуальні машини на базі тієї самої системної архітектури (VMs based on the same instruction set architecture, ISA) та віртуальні машини на базі іншої системної архітектури (VMs based on different ISA).

(Smith, Nair, 2005)



Рис. Б.12. Систематизація технологій віртуалізації: підхід Дж. Сміта та Р. Наїра (адаптована авторська схема)

У випадку процесових віртуальних машин до віртуальних машин на базі тієї самої системної архітектури віднесено всі багатозадачні системи (multiprogrammed systems) та динамічні двійкові оптимізатори на основі тієї самої ISA (same ISA dynamic binary optimizers), а до віртуальних машин на базі іншої системної архітектури – динамічні транслятори (dynamic translators) та віртуальні машини мов програмування високого рівня (high-level language VMs). У випадку системних віртуальних машин до віртуальних машин на базі тієї самої системної архітектури належать класичні системні віртуальні машини (classic system VMs) і гостьові віртуальні машини (hosted VMs), а до віртуальних машин на базі іншої системної архітектури – повносистемні віртуальні машини (whole system VMs) і спільно розроблені віртуальні машини (codesigned VMs).

Підхід авторів Ю. Лі (Yunfa Li), В. Лі (Wanqing Li), К. Цзян (Congfeng Jiang) [244] передбачає поділ технологій віртуалізації на повну віртуалізацію (full

virtualization), паравіртуалізацію (paravirtualization), апаратну віртуалізацію (hardware assisted virtualization), віртуалізація ресурсів (resource virtualization) – рис. Б.13.

(Li, Li, Jiang, 2010)

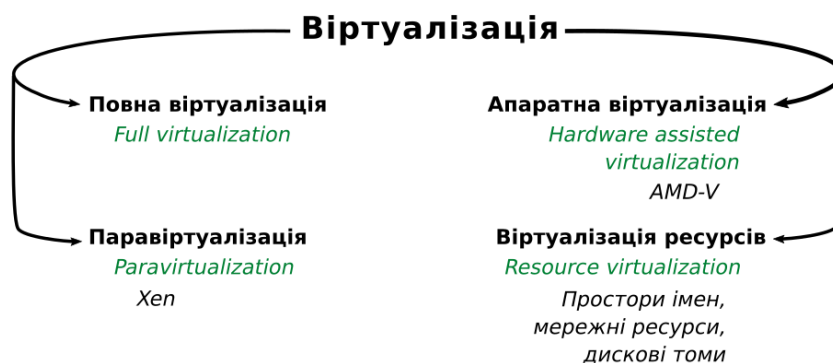


Рис. Б.13. Систематизація технологій віртуалізації: складено за підходом авторів Ю. Лі, В. Лі, К. Цзян

Як приклад паравіртуалізації автори вказують Xen, апаратної віртуалізації – AMD-V, віртуалізації ресурсів – простори імен, мережні ресурси, дискові томи.

Підхід С. М. Яшанова [193]. С. М. Яшанов виокремлює віртуалізацію платформ та віртуалізацію ресурсів, причому далі основну увагу у даній систематизації приділено передусім віртуалізації платформ (рис. Б.14).

(Яшанов, 2010)

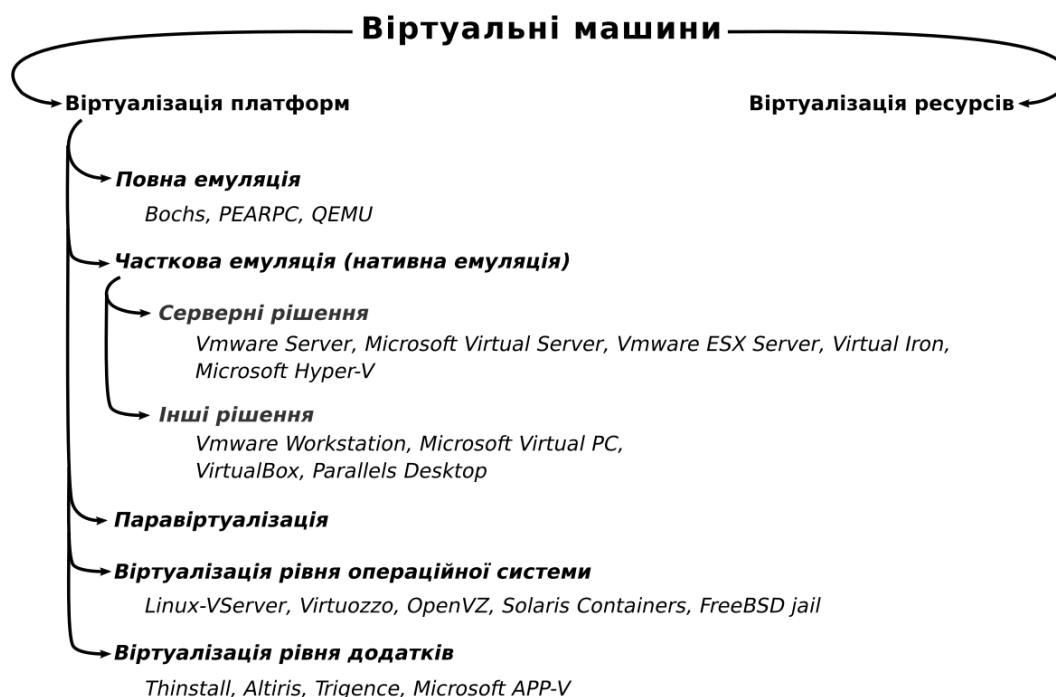


Рис. Б.14. Систематизація технологій віртуалізації: складено за підходом С. М Яшанова

До віртуалізації платформ автор відносить повну емуляцію (Bochs, PEARPC, QEMU), часткову або нативну, емуляцію (Vmware Workstation, Microsoft Virtual PC, VirtualBox, Parallels Desktop, окремо згадано серверні рішення: Vmware Server, Microsoft Virtual Server, Vmware ESX Server, Virtual Iron, Microsoft Hyper-V), паравіртуалізацію, віртуалізацію рівня ОС (Linux-VServer, Virtuozzo, OpenVZ, Solaris Containers, FreeBSD jail) та віртуалізацію рівня додатків (Thinstall, Altiris, Trigence, Microsoft App-V).

Додаток В.

ПОРІВНЯННЯ КУРСІВ З ОС У ЗВО УКРАЇНИ

Таблиця В.1

Порівняння курсів з ОС у ЗВО України: фактичні дані

№ п/п	Назва ЗВО	Спеціальності, для яких викладається курс з ОС	Назви курсу	Застосування засобів віртуалізації під час лабораторних робіт
1	Вінницький національний технічний університет [25]	121 «Інженерія програмного забезпечення», 122 «Комп'ютерні науки», 123 «Комп'ютерна інженерія», 125 «Кібербезпека», 126 «Інформаційні системи та технології», 151 «Автоматизація та комп'ютерно-інтегровані технології»	«Операційні системи», «Системне програмування», «Системне адміністрування», «Спеціалізовані операційні системи», «Мережні операційні системи», «Системне програмне забезпечення»	Інформація відсутня
2	Дніпропетровський національний університет імені Олеся Гончара [58]	121 «Інженерія програмного забезпечення», 122 «Комп'ютерні науки», 123 «Комп'ютерна інженерія», 125 «Кібербезпека», 113 «Прикладна математика»	«Операційні системи», «Unix-подібні операційні системи», «Системне програмування», «Програмування в операційних середовищах»	Інформація відсутня
3	Донецький національний університет ім. Василя Стуса [34]	122 «Комп'ютерні науки», 125 «Кібербезпека»	«Операційні системи», «Розробка додатків та адміністрування в ОС Linux»	Віртуальні машини [144]
4	Київський національний університет імені Тараса Шевченка [177; 178; 179]	121 «Інженерія програмного забезпечення», 122 «Комп'ютерні науки та інформаційні технології», 123 «Комп'ютерна інженерія», 125 «Кібербезпека», 126 «Інформаційні системи та технології» (факультет інформаційних технологій: 121, 122, 125, 126; факультет комп'ютерних наук та кібернетики: 121, 122; факультет радіофізики, електроніки та комп'ютерних систем: 123)	«Операційні системи», «Системне програмування», «Системне програмування та операційні системи»	Інформація відсутня
6	Криворізький державний педагогічний університет [82]	014.09 «Середня освіта (Інформатика)»	«Операційні системи та системне програмування»	Інформація відсутня

№ п/п	Назва ЗВО	Спеціальності, для яких викладається курс з ОС	Назви курсу	Застосування засобів віртуалізації під час лабораторних робіт
7	Львівський національний університет імені Івана Франка [90]	122 «Комп'ютерні науки», 126 «Інформаційні системи та технології», 113 «Прикладна математика», 014.09 «Середня освіта (Інформатика)» ; 2015 рік вступу - 6.040302 «Інформатика»	«Операційні системи», «Операційні системи та системне програмування»	До 2011 року – без віртуалізації (окремий комп'ютерний клас), 2011/2012 і 2012/2013 н. р. – сервер з Debian та OpenVZ, з 2012/2013 н. р. – сервер з Debian + Proxmox VE + OpenVZ та KVM [5]
8	Національний авіаційний університет [107]	121 «Інженерія програмного забезпечення», 122 «Комп'ютерні науки», 123 «Комп'ютерна інженерія», 125 «Кібербезпека», 113 «Прикладна математика», 151 «Автоматизація та комп'ютерно-інтегровані технології»	«Операційні системи», «Операційні системи та їх безпека», «Системне програмування», «Системне програмне забезпечення»	Інформація відсутня
9	Національний аерокосмічний університет ім. М. Є. Жуковського (ХАІ) [35]	121 «Інженерія програмного забезпечення», 122 «Комп'ютерні науки», 123 «Комп'ютерна інженерія», 125 «Кібербезпека», 126 «Інформаційні системи та технології», 113 «Прикладна математика», 151 «Автоматизація та комп'ютерно-інтегровані технології»	«Операційні системи», «Системне програмування та операційні системи»	Інформація щодо застосування засобів віртуалізації для курсу з ОС відсутня. У ЗВО створено хмарну інфраструктуру, й однією з проблем, запланованих до вирішення під час проектування цієї інфраструктури, є надання студентам адміністративного доступу до системи, що вивчається [135]
10	Національний педагогічний університет імені М. П. Драгоманова [108]	014.09 «Середня освіта (Інформатика)», 113 «Прикладна математика»	«Інформатика. Операційні системи»	Інформація відсутня
11	Національний технічний університет "Харківський політехнічний інститут" [109]	121 «Інженерія програмного забезпечення», 122 «Комп'ютерні науки», 123 «Комп'ютерна інженерія», 125 «Кібербезпека», 126 «Інформаційні системи та технології», 113 «Прикладна математика», 151 «Автоматизація та комп'ютерно-інтегровані технології»	«Операційні системи», «Системне програмування і операційні системи», «Операційні системи та системне програмування»	Інформація відсутня

№ п/п	Назва ЗВО	Спеціальності, для яких викладається курс з ОС	Назви курсу	Застосування засобів віртуалізації під час лабораторних робіт
12	Національний технічний університет «Дніпровська політехніка» [115]	121 «Інженерія програмного забезпечення», 122 «Комп'ютерні науки», 123 «Комп'ютерна інженерія», 125 «Кібербезпека», 151 «Автоматизація та комп'ютерно-інтегровані технології»	«Операційні системи»	Oracle Virtual Box. У ЗВО діє програма Microsoft Virtual Academy, яка може передбачати доступ до віртуальних машин у хмарі на базі Microsoft Azure (станом на 2016 рік) [134]
13	Національний університет "Львівська політехніка" [112]	121 «Інженерія програмного забезпечення», 122 «Комп'ютерні науки», 123 «Комп'ютерна інженерія», 125 «Кібербезпека»	«Операційні системи», «Системне програмування», «Операційні системи комп'ютерних мереж», «Операційні системи корпоративних мереж»	Інформація відсутня
14	Національний університет «Києво-Могилянська академія» [111]	121 «Інженерія програмного забезпечення», 122 «Комп'ютерні науки», 113 «Прикладна математика»	«Основи Unix систем», «Основи операційних систем»	Інформація відсутня
15	Національний університет біоресурсів і природокористування України [110]	121 «Інженерія програмного забезпечення», 122 «Комп'ютерні науки», 123 «Комп'ютерна інженерія», 125 «Кібербезпека»	«Операційні системи», «Операційні системи реального часу», «Операційні системи мобільних пристроїв», «Основи системного програмування», «Системне програмування», «Системне програмне забезпечення»	Інформація щодо застосування технологій віртуалізації безпосередньо для курсу з ОС відсутня. У ЗВО створено хмарну інфраструктуру [32]
16	НТУУ «Київський політехнічний інститут імені Ігоря Сікорського» (КПІ) [81]	121 «Інженерія програмного забезпечення», 122 «Комп'ютерні науки», 123 «Комп'ютерна інженерія», 125 «Кібербезпека», 126 «Інформаційні системи та технології», 113 «Прикладна математика»	«Операційні системи», «Системне програмування», «Системне програмування і операційні системи», «Операційна система UNIX», «Мережні операційні системи», «Системне програмне забезпечення»; Академія LPI (Linux Professional Institute)	Віртуальні машини (станом на 2014 рік – VirtualBox; станом на 2018 рік у ЗВО наявні Microsoft Hyper-V, VMware Workstation) [167]
17	Одеський національний політехнічний університет [117]	121 «Інженерія програмного забезпечення», 122 «Комп'ютерні науки», 123 «Комп'ютерна інженерія», 125 «Кібербезпека», 126 «Інформаційні системи та технології», 113 «Прикладна математика», 151 «Автоматизація та комп'ютерно-інтегровані технології»	«Операційні системи», «Системне програмування та системне програмне забезпечення», «Системне програмування», «Низькорівневе системне програмування»	Oracle VirtualBox [130]

№ п/п	Назва ЗВО	Спеціальності, для яких викладається курс з ОС	Назви курсу	Застосування засобів віртуалізації під час лабораторних робіт
18	Сумський державний університет [165]	122 «Комп'ютерні науки», 125 «Кібербезпека»; до 2016 року – 6.040302 «Інформатика»	«Мережні операційні системи», «Операційні системи», «Системне програмування та операційні системи»	Інформація відсутня
19	Тернопільський національний педагогічний університет імені Володимира Гнатюка [124]	014.09 «Середня освіта (Інформатика)»	«Операційні системи»	Віртуальні машини у корпоративній хмарі на базі платформи CloudStack [118]
20	Харківський національний економічний університет імені Семена Кузнеця [36]	121 «Інженерія програмного забезпечення», 122 «Комп'ютерні науки», 125 «Кібербезпека», 126 «Інформаційні системи та технології»	«Операційні системи»	Віртуальні машини [120]
21	Харківський національний університет радіоелектроніки [181]	121 «Інженерія програмного забезпечення», 122 «Комп'ютерні науки», 123 «Комп'ютерна інженерія», 125 «Кібербезпека», 126 «Інформаційні системи та технології»	«Операційні системи» (121, 125), «Операційні системи та системне програмування» (122), «Системне програмне забезпечення» (123), «Системне програмування» (123), «Системне програмування і операційні системи» (126)	Інформація відсутня
22	Чернівецький національний університет імені Юрія Федьковича [147]	121 «Інженерія програмного забезпечення», 122 «Комп'ютерні науки», 123 «Комп'ютерна інженерія», 125 «Кібербезпека», 126 «Інформаційні системи та технології», 014.09 «Середня освіта (Інформатика)»	«Операційні системи», «Системне програмування», «Мережні операційні системи»	Інформація відсутня
23	Чорноморський національний університет імені Петра Могили [184]	121 «Інженерія програмного забезпечення», 122 «Комп'ютерні науки», 123 «Комп'ютерна інженерія»	«Операційні системи», «Системне програмування», «Адміністрування та налагодження комп'ютерних систем»	Віддалений доступ (PuTTY) до віртуалізованого Debian Linux на сервері університету [119]

Додаток Г.

НАВЧАЛЬНА ТА РОБОЧА ПРОГРАМИ З ДИСЦИПЛІНИ «ОПЕРАЦІЙНІ СИСТЕМИ ТА СИСТЕМНЕ ПРОГРАМУВАННЯ»

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЖИТОМИРСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА

"ЗАТВЕРДЖУЮ"

Проректор з навчальної роботи

Л. М. Янович



[Signature]
06 2015 року

ОПЕРАЦІЙНІ СИСТЕМИ ТА СИСТЕМНЕ ПРОГРАМУВАННЯ

ПРОГРАМА

навчальної дисципліни варіативного циклу
підготовки за освітньо-кваліфікаційним рівнем бакалавра
спеціальності 6.040302 "Інформатика*"
фізико-математичний факультет

(шифр і назва спеціальності)

Житомир 2015

РОЗРОБНИК ПРОГРАМИ:

Головня О. С., асистент кафедри прикладної математики та інформатики
(науковий ступінь, звання, прізвище, ініціали, посада (з указанням кафедри))

Загацька Н. О., асистент кафедри прикладної математики та інформатики
(науковий ступінь, звання, прізвище, ініціали, посада (з указанням кафедри))

Навчальна програма затверджена на засіданні кафедри прикладної математики та інформатики

“23” 06 20 15 року, протокол № 12.

Зав. кафедрою проф. Сирій О.М.

Схвалено методичною комісією фізико-математичного факультету

“26” 06 20 15 року, протокол № 10.

Голова _____ доц. Франовський А. Ц.

Схвалено вченою радою Житомирського державного університету імені Івана Франка

“26” 06 20 15 року, протокол № 11.

Проректор з навчальної роботи Л. М. Янович

ПОЯСНЮВАЛЬНА ЗАПИСКА

Програму вивчення навчальної дисципліни “Операційні системи та системне програмування” складено відповідно до освітньо-професійної програми підготовки бакалаврів напряму 6.040302 Інформатика* галузі знань 0403 Системні науки та кібернетика.

Предметом вивчення навчальної дисципліни є операційні системи, їх будова, функціонування, особливості роботи з ними та програмування під них.

Міждисциплінарні зв'язки: “Програмування”, “Теорія програмування”, “Захист інформації у комп'ютерних системах”, “Комп'ютерні мережі та Інтернет”, “Технології програмування”.

Програма навчальної дисципліни складається з наступних змістових модулів:

1. Основи адміністрування сучасних операційних систем та створення програм для них.

1. Мета та завдання навчальної дисципліни

1.1. **Метою** викладання навчальної дисципліни “Операційні системи та системне програмування” є формування в майбутніх бакалаврів інформатики знань з теорії операційних систем, а також вироблення у них практичних навичок щодо їх застосування.

1.2. Основними **завданнями** вивчення дисципліни “Операційні системи та системне програмування” є:

- надання студентам необхідних теоретичних знань про будову та функціонування сучасних операційних систем;
- вироблення в студентів навичок адміністрування цих систем;
- навчання студентів створення системного програмного забезпечення під різні операційні системи.

1.3. Згідно з вимогами освітньо-кваліфікаційної програми студенти повинні знати:

- основні принципи організації, можливості та функціонування сучасних операційних систем;
- суть поняття віртуальної машини та технології віртуалізації, а також сфери застосування віртуалізації;
- головні тенденції сучасного ринку операційних систем;
- принципи створення сучасного системного програмного забезпечення.

вміти:

- здійснювати адміністрування ОС Windows та Linux;
- застосовувати віртуальні машини (гіпервізор VirtualBox);
- використовувати сучасні засоби розробки програмного забезпечення (GCC Compiler, IDE Code::Blocks);
- створювати системне програмне забезпечення для ОС Windows та Linux.

Навчальна дисципліна “Операційні системи та системне програмування” викладається на II курсі у II семестрі.

На вивчення навчальної дисципліни відводиться 54 години, 1,5 кредити ЄКТС.

Форма проведення занять: лекції, лабораторні роботи.

2. Інформаційний обсяг навчальної дисципліни

Змістовий модуль 1

ОСНОВИ АДМІНІСТРУВАННЯ СУЧАСНИХ ОПЕРАЦІЙНИХ СИСТЕМ ТА СТВОРЕННЯ ПРОГРАМ ДЛЯ НИХ

Тема 1. Початкові відомості про операційні системи

Початкові відомості про операційні системи. Поняття операційної системи (ОС). Історія розвитку ОС. Структура ОС. Класифікація ОС.

Тема 2. Процеси та потоки

Поняття процесу. Поняття потоку. Класифікація ОС за кількістю підтримуваних процесів та потоків. Стани процесів. Дескриптори процесів та потоків. Операції над процесами та потоками.

Тема 3. Пам'ять

Види пам'яті. Поняття віртуальної пам'яті. Логічна і фізична адресація. Підхід базового і межового реєстрів. Сегментна організація пам'яті. Сторінкова організація пам'яті. Сторінково-сегментна організація пам'яті. Підкачування. Заміщення сторінок.

Тема 4. Файлові системи

Файл та файлова система. Файлові атрибути. Каталоги. Операції над файлами та каталогами. Файлові системи сімейства FAT. Файлова система NTFS. Файлова система UFS. Файлові системи сімейства ext.

Тема 5. Безпека

Основні завдання системи безпеки. Аутентифікація та керування доступом. Аудит. Журналізація. Дискові квоти. Шифрування даних. Мережна безпека даних.

Тема 6. Планування

Планувальник. Алгоритм планування. Об'єкти планування. Вимоги до планування. Процеси, обмежені швидкістю роботи обчислень, та процеси, обмежені швидкістю роботи пристроїв введення-виведення. Алгоритми планування без витісненням та з витісненням. Планування у системах пакетної обробки. Планування в інтерактивних системах. Планування у системах реального часу. Особливості планування у Linux та Windows.

Тема 7. Міжпроцесова та міжпотокowa взаємодія

Взаємодія процесів. Синхронізація. Семафори. М'ютекси. Умовні змінні. Монітори. Блокування читання-запису. Бар'єри. Передача даних між процесами.

Тема 8. Віртуалізація

Поняття віртуалізації. Напрями та методи віртуалізації. Віртуалізація серверів. Віртуальні машини та віртуальні контейнери. Віртуальний хостинг. Хмарні технології.

3. Рекомендована література

Основна

1. Колисниченко Д.Н. Ubuntu Linux. Краткое руководство пользователя. – СПб.: БХВ-Петербург, 2007. – 304 с.: ил.
2. Операционная система Linux: курс лекций: учеб. пособие для студентов вузов, обучающихся по специальностям в области информационных технологий / Г. В. Курячий, К. А. Маслинский. – М.: Интернет-Ун-т Информ. Технологий, 2005. – 392 с. – (Серия "Основы информационных технологий").
3. Операційні системи: навч. посібник / М.Ф. Бондаренко, О.Г. Качко. – Х.: Компанія СМІТ, 2008. – 432 с.
4. Побегайло А.П. Системное программирование в Windows. – СПб.: БХВ-Петербург, 2006. – 1056 с.: ил.
5. Сетевые операционные системы / В.Г. Олифер, Н.А. Олифер. – СПб.: Питер, 2002. – 544 с.: ил.
6. Станек У.Р. Командная строка Microsoft Windows. Справочник администратора: Пер. с англ. – М., 2004. – 480 с.: ил.
7. Таненбаум Э. Современные операционные системы. 2-е изд. – СПб.: Питер, 2002. – 1040 с.: ил.
8. Хонейкатт Дж. Реестр Microsoft Windows XP. Справочник профессионала: Практ. пособ. / Пер. с англ. – М.: Издательство "СП ЭКОМ", 2003. – 656 с.: ил.
9. Шеховцов В.А. Операційні системи. – К.: Видавнича група BVH, 2008. – 576 с.: іл.

Додаткова

10. Белунцов В. Самоучитель пользователя Linux – М.: "ДЕСС КОМ", 2003. – 512 с.
11. Еаррет Д. Дж. Linux: основные команды. Карманный справочник / Пер. с англ. – М.: КУДИЦ-ОБРАЗ, 2005. – 288 с.
12. Кейслер С. Проектирование операционных систем для малых ЭВМ: Пер. с англ. – М.: Мир, 1986. – 680 с.
13. Костромин В.А. Самоучитель Linux для пользователя. – СПб.: БХВ-Петербург, 2003. – 672 с.
14. Лав Р. Разработка ядра Linux, 2е издание: Пер. с англ. – М.: "Вильямс", 2006. – 448 с.
15. Митчел М., Оулдем Дж., Самьюэл А. Программирование для Linux. Профессиональный подход.: Пер. с англ. – М.: "Вильямс", 2003. – 288 с.
16. Наик Д. Системы хранения данных в Windows: Пер. с англ. – М.: Издательский дом "Вильямс", 2005. – 432 с.: ил.
17. Смит Р.В. Сетевые средства Linux.: Пер. с англ. – М.: "Вильямс", 2003. – 672 с.
18. Таненбаум Э., Вудхалл А. Операционные системы: разработка и реализация. Классика CS. – СПб.: Питер, 2006. – 576 с.
19. Тейсли Д. Linux и UNIX: программирование в shell. Руководство разработчика.: Пер. с англ. – К.: Издательская группа BHV, 2001. – 464 с.
20. Шоу А. Логическое проектирование операционных систем: Пер. С 288нгл.. – М.: Мир, 1981. – 360 с.

16. Інформаційні ресурси

4. Форма підсумкового контролю успішності

Екзамен.

Вимоги до екзамену:

1. Володіння теоретичним матеріалом з курсу “Операційні системи та системне програмування”.
2. Виконання усіх завдань лабораторних робіт, передбачені робочою програмою та інструктивно-методичними матеріалами.
3. Пройдено перевірку рівня засвоєння знань з тем, що виносились на самостійне опрацювання, на консультації у викладача.

5. Методи контролю успішності навчання

Поточний тестовий контроль, лабораторні роботи, підсумкові модульні роботи, перевірка рівня засвоєння тем, винесених на самостійне опрацювання, екзамен.

ЖИТОМИРСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА
ФІЗИКО-МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ
КАФЕДРА ПРИКЛАДНОЇ МАТЕМАТИКИ ТА ІНФОРМАТИКИ

"ЗАТВЕРДЖУЮ"

Проректор з навчальної роботи

Л. М. Янович



06 2015 року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Операційні системи та системне програмування

напрямок підготовки 6.040302 "Інформатика*"

Житомир 2015

Робоча програма **“Операційні системи та системне програмування”** для студентів напряму підготовки 6.040302 Інформатика*

РОЗРОБНИК ПРОГРАМИ:

Головня О. С., асистент кафедри прикладної математики та інформатики

(науковий ступінь, звання, прізвище, ініціали, посада (з указанням кафедри))

Загацька Н. О., асистент кафедри прикладної математики та інформатики

(науковий ступінь, звання, прізвище, ініціали, посада (з указанням кафедри))

Навчальна програма затверджена на засіданні кафедри прикладної математики та інформатики

“23” 06 20 15 року, протокол № 12.

Зав. кафедрою проф. Стрип О.М.

Схвалено методичною комісією фізико-математичного факультету

“26” 06 20 15 року, протокол № 10.

Голова _____ доц. Франовський А. Ц.

Схвалено вченою радою Житомирського державного університету імені Івана Франка

“26” 06 20 15 року, протокол № 11.

Проректор з навчальної роботи Л. М. Янович

(с) Головня О. С., 2015 рік

(с) Загацька Н. О., 2015 рік

1. Опис навчальної дисципліни

Найменування показників	Галузь знань, напрям підготовки, освітньо-кваліфікаційний рівень	Характеристика навчальної дисципліни	
		Денна форма навчання	Заочна форма навчання
Кількість кредитів — 1,5	Галузь знань 0403 Системні науки та кібернетика	За вибором	
Модулів — 1	Напрямок підготовки 6.040302 Інформатика*	Рік підготовки	
Змістових модулів — 1		2-й	2-й
Загальна кількість годин — 54		Семестр	
		II	II
		Лекції:	
		18 год	4 год
Тижневих годин для денної форми навчання: аудиторних — 2,12 самостійної роботи студента — 1	Освітньо-кваліфікаційний рівень: бакалавр	Практичні:	
		—	—
		Лабораторні:	
		18 год	10 год
		Самостійна робота:	
		18 год	40 год
		Індивідуальні завдання:	
		—	—
		Вид контролю:	
		Екзамен	Екзамен

Співвідношення кількості годин аудиторних занять до самостійної та індивідуальної роботи становить (%):

для денної форми навчання — 67% : 33%

для заочної форми навчання — 26% : 74%

2. Мета та завдання навчальної дисципліни

1.1. **Метою** викладання навчальної дисципліни “Операційні системи та системне програмування” є формуванні в майбутніх бакалаврів інформатики знань з теорії операційних систем, а також вироблення у них практичних навичок щодо їх застосування.

1.2. Основними **завданнями** вивчення дисципліни “Операційні системи та системне програмування” є:

- надання студентам необхідних теоретичних знань про будову та функціонування сучасних операційних систем;
- вироблення в студентів навичок адміністрування цих систем;
- навчання студентів створення системного програмного забезпечення під різні операційні системи.

1.3. Згідно з вимогами освітньо-кваліфікаційної програми студенти повинні знати:

- основні принципи організації, можливості та функціонування сучасних операційних систем;
- суть поняття віртуальної машини та технології віртуалізації, а також сфери застосування віртуалізації;
- головні тенденції сучасного ринку операційних систем;
- принципи створення сучасного системного програмного забезпечення.

вміти:

- здійснювати адміністрування ОС Linux (додатково: ОС Windows);
- застосовувати віртуальні машини (гіпервізор VirtualBox);
- використовувати сучасні засоби розробки програмного забезпечення (GCC Compiler, IDE Code::Blocks);
- створювати системне програмне забезпечення для ОС Linux (додатково: ОС Windows).

3. Програма навчальної дисципліни

Змістовий модуль 1

ОСНОВИ АДМІНІСТРУВАННЯ СУЧАСНИХ ОПЕРАЦІЙНИХ СИСТЕМ ТА СТВОРЕННЯ ПРОГРАМ ДЛЯ НИХ

Тема 1. Початкові відомості про операційні системи

Початкові відомості про операційні системи. Поняття операційної системи (ОС). Історія розвитку ОС. Структура ОС. Класифікація ОС.

Тема 2. Процеси та потоки

Поняття процесу. Поняття потоку. Класифікація ОС за кількістю підтримуваних процесів та потоків. Стани процесів. Дескриптори процесів та потоків. Операції над процесами та потоками.

Тема 3. Пам'ять

Види пам'яті. Поняття віртуальної пам'яті. Логічна і фізична адресація. Підхід базового і межового реєстрів. Сегментна організація пам'яті. Сторінкова організація пам'яті. Сторінково-сегментна організація пам'яті. Підкачування. Заміщення сторінок.

Тема 4. Файлові системи

Файл та файлова система. Файлові атрибути. Каталоги. Операції над файлами та каталогами. Файлові системи сімейства FAT. Файлова система NTFS. Файлова система UFS. Файлові системи сімейства ext.

Тема 5. Безпека

Основні завдання системи безпеки. Аутентифікація та керування доступом. Аудит. Журналізація. Дискові квоти. Шифрування даних. Мережна безпека даних.

Тема 6. Планування

Планувальник. Алгоритм планування. Об'єкти планування. Вимоги до планування. Процеси, обмежені швидкістю роботи обчислень, та процеси, обмежені швидкістю роботи пристроїв введення-виведення. Алгоритми планування без витісненням та з витісненням. Планування у системах пакетної обробки. Планування в інтерактивних системах. Планування у системах реального часу. Особливості планування у Linux та Windows.

Тема 7. Міжпроцесова та міжпотоква взаємодія

Взаємодія процесів. Синхронізація. Семафори. М'ютекси. Умовні змінні. Монітори. Блокування читання-запису. Бар'єри. Передача даних між процесами.

Тема 8. Віртуалізація

Поняття віртуалізації. Напрями та методи віртуалізації. Віртуалізація серверів. Віртуальні машини та віртуальні контейнери. Віртуальний хостинг. Хмарні технології.

4. Структура навчальної дисципліни

№	Теми	Кількість годин							
		Денна				Заочна			
		Всього	У тому числі			Всього	У тому числі		
			Лекції	Лаб.	Сам. роб.		Лекції	Лаб.	Сам. роб.
Модуль 1. Основи адміністрування сучасних операційних систем та створення програм для них									
1	Початкові відомості про операційні системи	8	2	4	2	8	1	2	5
2	Процеси та потоки	9	3	4	2	11	2	4	5
3	Пам'ять	6	2	2	2	6	1	0	5
4	Файлові системи	9	2	4	3	7	0	2	5
5	Безпека	9	3	4	2	7	0	2	5
6	Планування	5	2	0	3	5	0	0	5
7	Міжпроцесова та міжпотоківна взаємодія	4	2	0	2	5	0	0	5
8	Віртуалізація	4	2	0	2	5	0	0	5
Всього за I модуль		54	18	18	18	54	4	10	40
Всього з дисципліни		54	18	18	18	54	4	10	40

5. Теми лекційних занять

№	Тема	Кількість годин
1	Початкові відомості про операційні системи	2
2	Процеси та потоки	3
3	Пам'ять	2
4	Файлові системи	2
5	Безпека	3
6	Планування	2
7	Міжпроцесова та міжпотоківна взаємодія	2
8	Віртуалізація	2
Разом:		18

Теми лекційних занять (заочна форма навчання)

№	Тема	Кількість годин
1	Початкові відомості про операційні системи	1
2	Процеси та потоки	2
3	Пам'ять	1
4	Файлові системи	0
5	Безпека	0
6	Планування	0
7	Міжпроцесова та міжпотоківна взаємодія	0
8	Віртуалізація	0
Разом:		4

6. Теми семінарських занять

Не передбачені навчальним планом.

7. Теми практичних занять

Не передбачені навчальним планом.

8. Теми лабораторних занять (денна форма навчання)

№	Тема	Кількість годин
1	Знайомство з Linux	2
2	Налаштування мережного з'єднання у Linux	2
3	Встановлення ПЗ у Linux	2
4	Керування процесами у Linux	2
5	Програмування процесів та потоків у Linux	2
6	Моніторинг використання пам'яті та процесорного часу у Linux	2
7	Робота з файлами у Linux	4
8	Користувачі та групи у Linux	2
Разом:		18

Теми лабораторних занять (заочна форма навчання)

№	Тема	Кількість годин
1	Знайомство з Linux	2
2	Встановлення ПЗ у Linux	2
3	Керування процесами у Linux	2
4	Програмування процесів та потоків у Linux	1
5	Робота з файлами у Linux	1
6	Користувачі та групи у Linux	1
7	Користувачі та групи у Windows	1
Разом:		10

9. Самостійна робота (денна форма навчання)

Тема	Індивідуальні завдання	Кількість годин	Література, №
Модуль 1. Основи адміністрування сучасних операційних систем та створення програм для них			
Початкові відомості про операційні системи	1. Операційна система MINIX. 2. Операційна система MULTIX. 3. Основи роботи з віртуальними машинами 4. Встановлення Linux	2	[3], [5], [7],
Процеси та потоки	1. Порівняння процесів та потоків у Windows та Linux.	2	[7], [3], [9]
Пам'ять	1. Пробуксовування.	2	[7], [9]
Файлові системи	1. Продуктивність файлових систем. 2. Особливості кешування у Windows.	3	[7], [3]
Безпека	1. Шифровані файлові системи. 2. Атаки і боротьба з ними.	2	[7], [9], [5]
Планування	1. Деталі планування у Windows 2. Деталі планування у Linux	3	[7]
Міжпроцесова та міжпоточкова взаємодія	1. Стандартні сигнали стандарту POSIX 2. Ф'ютекси. 3. Порівняння реалізацій планування у Windows та Linux. 4. Обробка винятків в об'єктно-орієнтованих мовах програмування.	2	[2], [7], [9], [4]
Віртуалізація	1. Серверні засоби в Ubuntu Linux 2. Організація веб-сервера на базі Ubuntu Linux та VirtualBox.	2	[1]
Разом		18	

Самостійна робота (заочна форма навчання)

Тема	Індивідуальні завдання	Кількість годин	Література, №
Модуль 1. Основи адміністрування сучасних операційних систем та створення програм для них			
Початкові відомості про операційні системи	1. Операційна система MINIX. 2. Операційна система MULTIX. 3. Основи роботи з віртуальними машинами 4. Встановлення Linux	5	[3], [5], [7],
Процеси та потоки	1. Порівняння процесів та потоків у Windows та Linux.	5	[7], [3], [9]
Керування пам'яттю	1. Пробуксовування. 2. Моніторинг використання пам'яті та процесорного часу засобами Windows та Linux.	5	[7], [9]
Файлові системи	1. Продуктивність файлових систем. 2. Особливості кешування у Windows.	5	[7], [3]
Безпека	1. Шифровані файлові системи. 2. Атаки і боротьба з ними.	5	[7], [9], [5]
Планування	1. Деталі планування у Windows 2. Деталі планування у Linux	5	[7]
Міжпроцесова та міжпоточкова взаємодія	1. Стандартні сигнали стандарту POSIX 2. Ф'ютекси. 3. Порівняння реалізацій планування у Windows та Linux. 4. Обробка винятків в об'єктно-орієнтованих мовах програмування.	5	[2], [7], [9], [4]
Віртуалізація	1. Серверні засоби в Ubuntu Linux 2. Організація веб-сервера на базі Ubuntu Linux та VirtualBox.	5	[1]
Разом		40	

10. Індивідуальні завдання

Не передбачені навчальним планом.

11. Методи навчання

Лекція, лабораторні роботи, бесіда, дедуктивно-індуктивні, проблемно-пошукові, самостійна робота.

12. Методи контролю

Поточний тестовий контроль, лабораторні роботи, підсумкові модульні роботи, перевірка рівня засвоєння тем, винесених на самостійне опрацювання, екзамен.

13. Розподіл балів, які отримують студенти

ТИП КОНТРОЛЮ	ФОРМИ КОНТРОЛЮ	КРИТЕРІЇ ОЦІНЮВАННЯ
Поточний	Лабораторна робота	1. Присутність студента на занятті під час виконання лабораторної роботи оцінюється в 1 бал. 2. Виконана та вчасно захищена лабораторна робота оцінюється в 99 балів. За невчасно захищену лабораторну роботу знімається 10 балів перший раз, 20 – другий і т.д. Загальна можлива набрана сума балів за лабораторну роботу – 100. 3. Якщо під час захисту роботи студент виявив добрі та відмінні теоретичні знання або результати самостійної роботи та зразково і творчо виконав практичну складову роботи, то роботу може бути додатково оцінено у 5-15 балів.
	Тести	Тести оцінюються комп'ютерною програмою, максимально можлива сума балів – 100.
Підсумковий	Аудиторна модульна контрольна робота	Аудиторна модульна контрольна робота оцінюється максимум в 100 балів.
	Іспит	1. Загальний бал виводиться середнім балом усіх оцінок (включаючи нулі). 2. Студент допускається до складання іспиту, якщо він має за роботу під час навчання не менше 31 балу. 3. Якщо загальна сума балів за результатами навчання перевищує 60, то студент за його бажанням та згодою викладача може отримати відповідну підсумкову оцінку без складання іспиту. 4. Іспит вважається складеним і ставиться відповідна оцінка, у випадку якщо студент набрав від 60 до 100 балів. «відмінно» - від 90 до 100 балів «добре» - від 74 до 89 балів «задовільно» - від 60 до 73 балів.

14. Методичне забезпечення

1. Навчальна програма з курсу.
2. Робоча програма з курсу.
3. Інструктивно-методичні матеріали з курсу.
4. Опорні конспекти лекцій, ілюстративні матеріали.
5. Головня О. С. Операційні системи та системне програмування: Методичний посібник для студ. вищих навч. закл. — Житомир: Видавництво ЖДУ, 2013. — 211 с., іл.

15. Рекомендована література

Основна

1. Колисниченко Д.Н. Ubuntu Linux. Краткое руководство пользователя. – СПб.: БХВ-Петербург, 2007. – 304 с.: ил.
2. Операционная система Linux: курс лекций: учеб. пособие для студентов вузов, обучающихся по специальностям в области информационных технологий / Г. В. Курячий, К. А. Маслинский. – М.: Интернет-Ун-т Информ. Технологий, 2005. – 392 с. – (Серия "Основы информационных технологий").
3. Операційні системи: навч. посібник / М.Ф. Бондаренко, О.Г. Качко. – Х.: Компанія СМІТ, 2008. – 432 с.
4. Побегайло А.П. Системное программирование в Windows. – СПб.: БХВ-Петербург, 2006. – 1056 с.: ил.
5. Сетевые операционные системы / В.Г. Олифер, Н.А. Олифер. – СПб.: Питер, 2002. – 544 с.: ил.
6. Станек У.Р. Командная строка Microsoft Windows. Справочник администратора: Пер. с англ. – М., 2004. – 480 с.: ил.
7. Таненбаум Э. Современные операционные системы. 2-е изд. – СПб.: Питер, 2002. – 1040 с.: ил.
8. Хонейкатт Дж. Реестр Microsoft Windows XP. Справочник профессионала: Практ. пособ. / Пер. с англ. – М.: Издательство "СП ЭКОМ", 2003. – 656 с.: ил.
9. Шеховцов В.А. Операційні системи. – К.: Видавнича група BVH, 2008. – 576 с.: іл.

Додаткова

10. Белунцов В. Самоучитель пользователя Linux – М.: "ДЕСС КОМ", 2003. – 512 с.
11. Еаррет Д. Дж. Linux: основные команды. Карманный справочник / Пер. с англ. – М.: КУДИЦ-ОБРАЗ, 2005. – 288 с.
12. Кейслер С. Проектирование операционных систем для малых ЭВМ: Пер. с англ. – М.: Мир, 1986. – 680 с.
13. Костромин В.А. Самоучитель Linux для пользователя. – СПб.: БХВ-Петербург, 2003. – 672 с.
14. Лав Р. Разработка ядра Linux, 2е издание: Пер. с англ. – М.: "Вильямс", 2006. – 448 с.
15. Митчел М., Оулдем Дж., Самьюэл А. Программирование для Linux. Профессиональный подход.: Пер. с англ. – М.: "Вильямс", 2003. – 288 с.
16. Наик Д. Системы хранения данных в Windows: Пер. с англ. – М.: Издательский дом "Вильямс", 2005. – 432 с.: ил.
17. Смит Р.В. Сетевые средства Linux.: Пер. с англ. – М.: "Вильямс", 2003. – 672 с.
18. Таненбаум Э., Вудхалл А. Операционные системы: разработка и реализация. Классика CS. – СПб.: Питер, 2006. – 576 с.
19. Тейсли Д. Linux и UNIX: программирование в shell. Руководство разработчика.: Пер. с англ. – К.: Издательская группа BVH, 2001. – 464 с.
20. Шоу А. Логическое проектирование операционных систем: Пер. С 300нгл.. – М.: Мир, 1981. – 360 с.

16. Інформаційні ресурси

1. ZDU PROJECT. Навчально-методичні матеріали. Операційні системи та системне програмування. — Режим доступу:
<http://project.zu.edu.ua/BookThisProject.php?project=%CE%EF%E5%F0%E0%F6%B3%E9%ED%B3%20%F1%E8%F1%F2%E5%EC%E8%20%F2%E0%20%F1%E8%F1%F2%E5%EC%ED%E5%20%EF%F0%EE%E3%F0%E0%EC%F3%E2%E0%ED%ED%FF>
2. Stack Overflow – Where Developers Learn, Share, & Build Careers. – Access mode:
<https://stackoverflow.com/>
3. Microsoft Source Developer Networking (MSDN) Library — Режим доступу:

- <http://msdn.microsoft.com/en-us/library/ms123401.aspx>
4. Проект OpenNet: Интерактивная система просмотра системных руководств (man-ов). – <http://www.opennet.ru/man.shtml?topic=&russian=0&category=&submit=%F0%CF%CB%C1%DA%C1%D4%D8+man>

Додаток Д.
ІНФОРМАТИЧНІ КОМПЕТЕНТНОСТІ БАКАЛАВРІВ ІНФОРМАТИКИ
У РІЗНИХ ДЖЕРЕЛАХ

Таблиця Д.1

**Встановлення відповідності між визначеними у різних джерелах інформатичними компетентностями
бакалаврів інформатики щодо ОС**

Спільні характеристики	Компетентності бакалаврів інформатики		
	Компетентності майбутніх вчителів інформатики (Спірін О. М., 2007)	Компетентності з галузевого стандарту для напрямку підготовки 6.040302 «Інформатика», 2010	Компетентності / програмні результати навчання з проекту галузевого стандарту для спеціальності 014.09 «Середня освіта (Інформатика)», 2017
Знання історії розвитку	Вміти логічно і послідовно подати засвоєні знання теоретичних, алгоритмічних основ інформатики та <u>історії її розвитку</u>	Вміти аналізувати <u>основні етапи розвитку інформаційних технологій</u> в історії цивілізації	—
Знання теоретичних основ	Вміти логічно і послідовно подати засвоєні знання <u>теоретичних, алгоритмічних основ інформатики</u> та історії її розвитку. Демонструвати знання <u>основ інформаційних систем</u> (архітектура, апаратне і програмне забезпечення ЕОМ, локальні та глобальні комп'ютерні мережі, мультимедіа тощо)	Володіти уявленнями про <u>логічні основи комп'ютерних систем</u> .	Здатність використовувати систематизовані <u>теоретичні та практичні знання наукових фактів, концепцій, теорій, принципів і методів інформатики</u> при вирішенні професійних завдань

Спільні характеристики	Компетентності бакалаврів інформатики		
	Компетентності майбутніх вчителів інформатики (Спірін О. М., 2007)	Компетентності з галузевого стандарту для напряму підготовки 6.040302 «Інформатика», 2010	Компетентності / програмні результати навчання з проекту галузевого стандарту для спеціальності 014.09 «Середня освіта (Інформатика)», 2017
Знання основних класифікацій	Демонструвати знання <u>основ інформаційних систем (архітектура, апаратне і програмне забезпечення ЕОМ, локальні та глобальні комп'ютерні мережі, мультимедіа тощо)</u>	Вміти <u>класифікувати операційні системи</u> та операційні оболонки, визначати їх склад	–
Здатність адмініструвати систему (інтерфейс ОС, мережні налаштування, встановлення ПЗ, керування процесами, потоками, моніторинг використання пам'яті, робота з файлами, налаштування системи безпеки)	Демонструвати знання <u>основ інформаційних систем (архітектура, апаратне і програмне забезпечення ЕОМ, локальні та глобальні комп'ютерні мережі, мультимедіа тощо)</u> . Вміти розв'язувати типові прикладні математичні, статистичні й соціально-економічні задачі, <u>ефективно використовувати сучасне системне (операційні системи; системи процедурного, функціонального, логічного, об'єктно-орієнтованого програмування; мови Інтернет-публікацій; сервісні програми)</u> та прикладне (загальне та спеціальне) програмне забезпечення ЕОМ. Уміти <u>встановлювати, налагоджувати</u> сучасні версії операційних систем, поширене прикладне та спеціальне програмне забезпечення ПК. Мати уявлення про <u>інформаційну безпеку</u> та вміти здійснювати елементарний <u>захист даних</u> на електронних носіях	Володіти досконалими програмістськими вміннями та <u>навичками користувача</u> персонального комп'ютера. Володіти основними прийомами <u>застосування системного програмного забезпечення</u> інформаційної системи. Вміти <u>інсталювати програмні засоби</u> . Вміти застосовувати основні функції операційної системи для роботи з <u>файлами та каталогами</u> . Вміти застосовувати <u>сервісне програмне забезпечення</u> (програми-архіватори, антивірусні програми, програми-утиліти тощо). Вміти <u>працювати з дисками</u> (форматувати та діагностувати диски, відновлювати дані тощо).	Здатність розробляти алгоритми розв'язування задач з інформатики, аналізувати складність й ефективність алгоритмів; реалізовувати алгоритми мовами програмування; <u>обирати та застосовувати програмне забезпечення для розв'язання прикладних задач</u>

Спільні характеристики	Компетентності бакалаврів інформатики		
	Компетентності майбутніх вчителів інформатики (Спірін О. М., 2007)	Компетентності з галузевого стандарту для напряму підготовки 6.040302 «Інформатика», 2010	Компетентності / програмні результати навчання з проекту галузевого стандарту для спеціальності 014.09 «Середня освіта (Інформатика)», 2017
Здатність до самонавчання	Вміти самостійно здійснювати <u>пошук та аналіз необхідних повідомлень</u> у контексті розвитку предметної галузі інформатики	Вміти користуватися системою меню та послугами <u>вбудованої системи допомоги та довідкової системи</u> Вміти здійснювати <u>пошук інформаційних джерел</u> із застосуванням інформаційних ресурсів мережі Інтернет	Здатність до <u>самостійного вивчення нових питань інформатики</u> та методики навчання інформатики; <u>інтегрувати знання</u> , здійснювати аналіз і порівняння педагогічних технологій, застосовувати логічні принципи побудови гіпотез і доведень
Вміння створювати системне програмне забезпечення	Демонструвати <u>розвинуте програмування</u> щонайменше однією з сучасних об'єктно-орієнтованих мов, наприклад C++	Володіти досконалими <u>програмістськими вміннями</u> та навичками користувача персонального комп'ютера. Вміти складати й реалізовувати <u>алгоритми з різними типами даних</u> . Вміти <u>вводити, налагоджувати та тестувати програми</u> на ПК. Вміти <u>складати програми</u> для розв'язування типових навчальних задач.	Знати <u>методи розробки та дослідження алгоритмів розв'язування задач з інформатики</u> , знати методики оцінювання ефективності алгоритмів; <u>володіти мовами програмування</u> різних видів, розуміти їх переваги для розв'язання базових задач інформатики. Здатність <u>розробляти алгоритми розв'язування задач з інформатики</u> , аналізувати складність й ефективність алгоритмів; <u>реалізовувати алгоритми мовами програмування</u> ; обирати та застосовувати програмне забезпечення для розв'язання прикладних задач.

Додаток Е.**ОПИС ГАЛУЗІ ЗНАНЬ «ОПЕРАЦІЙНІ СИСТЕМИ»:
ФРАГМЕНТ З КЕРІВНИХ ПРИНЦИПІВ РОЗРОБКИ НАВЧАЛЬНИХ
ПРОГРАМ COMPUTER SCIENCE CURRICULA 2013 (CS2013)**

Наступний фрагмент перекладено за документом [213, с. 135-141].

Операційні системи (ОС) / Operating Systems (OS)

Операційна система визначає абстракції для апаратного забезпечення та керує розподілом ресурсів між користувачами комп'ютера. Теми з цієї галузі знань пояснюють найбільш базові відомості про операційні системи у сенсі взаємодії операційної системи з мережами, відмінності між режимом ядра та режимом користувача, вироблення основних підходів до проектування і розробки операційних систем. Цю галузь знань структуровано таким чином, щоб бути взаємодоповнюваною з такими галузями, як «Основи систем» (System Fundamentals, SF), «Мережі та зв'язок» (Networking and Communication, NC), «Інформаційна безпека та захист» (Information Assurance and Security, IAS) і «Паралельні та розподілені обчислення» (Parallel and Distributed Computing, PD). Галузі знань «Основи систем» та «Інформаційна безпека та захист» нові і передбачають розгляд ряду особливостей, притаманних сучасним систем. Наприклад, галузь знань «Основи систем» містить питання, пов'язані з продуктивністю, віртуалізацією та ізоляцією, виділенням ресурсів, плануванням; галузь знань «Паралельні та розподілені обчислення» – основи паралелізму; галузь знань «Інформаційна безпека та захист» – детальний розгляд проблем безпеки і судової експертизи у справах про інформаційні злочини. У багатьох випадках курси з операційних систем можуть охоплювати матеріал з усіх цих галузей знань.

ОС. Операційні системи

(нормативні змістові модулі 1 рівня – 4 год.;

нормативні змістові модулі 2 рівня – 11 год.)

	Години для нормативних змістових модулів I рівня (Core Tier-1 hours)	Години для нормативних змістових модулів II рівня (Core Tier-2 hours)	Вибірковий змістовий модуль (Elective), так/ні
ОС/Огляд операційних систем	2		
ОС/Основні принципи роботи операційної системи	2		ні
ОС/Паралелізм		3	ні
ОС/Планування та диспетчеризація		3	ні
ОС/Керування пам'яттю		3	ні
ОС/Безпека і захист		2	ні
ОС/Віртуальні машини			так
ОС/Керування пристроями			так
ОС/Файлові системи			так
ОС/Системи реального часу та вбудовані системи			так
ОС/Відмовостійкість			так
ОС/Оцінювання продуктивності системи			так

ОС/Огляд операційних систем

[Нормативний 1 рівня, 2 год.]

Питання:

- Роль і призначення операційних систем
- Функціонування типової операційної системи
- Механізми підтримки клієнт-серверної моделі, мобільних пристроїв
- Особливості проектування (ефективність, надійність, гнучкість, портативність, безпека, сумісність)
- Вплив технологій безпеки, мережних та мультимедійних технологій, віконних систем

Результати навчання:

- пояснювати цілі та функції сучасних операційних систем
[Ознайомленість]
- аналізувати спадковість тенденцій у проектуванні операційних систем
[Застосування]
- описувати функції сучасної операційної системи з точки зору зручності, ефективності і перспектив для вдосконалення
[Ознайомленість]

- обговорювати мережні та розподілені операційні системи та клієнт-серверну архітектуру та їхні відмінності від однокористувацьких операційних систем [Ознайомленість]
- визначати потенційні загрози для операційних систем та засоби безпеки для захисту від них [Ознайомленість]

ОС/Основні принципи роботи операційної системи

[Нормативний 1 рівня, 2 год.]

Питання:

- Класифікація операційних систем за архітектурою (монолітні, багаторівневі, модульні, мікроядра)
- Абстракції, процеси і ресурси
- Поняття інтерфейсів прикладного програмування (API)
- Еволюція технологій апаратного/програмного забезпечення та потреб програмних засобів
- Організація роботи пристроїв
- Переривання: методи та реалізація
- Поняття режиму користувача та режиму ядра, перемикання у режим ядра

Результати навчання:

- пояснювати поняття логічного рівня [Ознайомленість]
- пояснювати переваги побудови абстрактних рівнів, впорядкованих ієрархічно [Ознайомленість]
- пояснювати цінність API та міжплатформного програмного забезпечення [Володіння]
- описувати, як обчислювальні ресурси використовуються прикладними програмами, і як системне програмне забезпечення керує цими ресурсами [Ознайомленість]
- порівнювати режим ядра та режим користувача в операційній системі [Застосування]
- обговорювати переваги та недоліки залучення механізму обробки переривань [Ознайомленість]
- пояснювати, як відбувається використання списку пристроїв та черг введення-виведення драйверів [Ознайомленість]

ОС/Паралелізм

[Нормативний 2 рівня, 3 год.]

Питання:

- Стани і діаграми станів (перехресне посилання: Основи систем/Стани і машини станів)
- Структури (списки процесів у стані готовності, блоки керування процесом тощо)

- Диспетчеризація та перемикання контекстів
- Роль переривань
- Організація атомарного доступу до об'єктів ОС
- Реалізація синхронізаційних примітивів
- Особливості роботи багатопроцесорних систем (циклічне блокування, повторний вхід)

Результати навчання:

- обґрунтовувати потребу в паралелізмі у межах операційної системи [Ознайомленість]
- демонструвати потенційні проблеми, що можуть виникати під час виконання паралельних операцій багатьма окремими задачами [Використання]
- окреслювати коло механізмів, які можуть бути застосовані на рівні операційної системи для реалізації паралелізму та пояснювати переваги кожного з цих механізмів [Ознайомленість]
- описувати різні стани, у яких може перебувати задача, та структури даних, необхідні для підтримки керування багатьма задачами [Ознайомленість]
- окреслювати коло технологій синхронізації в операційній системі (наприклад, описувати, як застосовувати семафори за допомогою примітивів ОС) [Ознайомленість]
- перераховувати причини для використання переривань, диспетчеризації та перемикання контекстів з метою підтримки паралелізму в операційній системі [Ознайомленість]
- створювати діаграми станів та переходів для простих проблемних областей [Застосування]

ОС/Планування та диспетчеризація

[Нормативний 2 рівня, 3 год.]

Питання:

- Планування з витісненням та без витіснення (перехресне посилання: Основи систем/Виділення ресурсів і планування, Паралельні і розподілені обчислення/Продуктивність паралельних обчислень)
- Планувальники і політики планування (перехресне посилання: Основи систем/Виділення ресурсів і планування, Паралельні і розподілені обчислення/Продуктивність паралельних обчислень)
- Процеси та потоки (Основи систем/Комп'ютерні парадигми)
- Дедлайни й особливості планування у системах реального часу

Результати навчання:

- порівнювати загальні алгоритми планування з витісненням та алгоритми планування без витіснення для планування задач в

- операційній системі, зокрема планування на основі пріоритетів, порівняння продуктивності, справедливого планування [Застосування]
- описувати взаємозв'язки між алгоритмами планування та областями їх застосування [Ознайомленість]
 - обговорювати типи планування роботи процесора, такі як короткострокове, середньострокове, довгострокове та введення-виведення [Ознайомленість]
 - пояснювати відмінність між процесами та потоками [Застосування]
 - порівнювати статичний та динамічний підходи до планування у системах реального часу [Застосування]
 - обговорювати питання потреби переривань та дедлайнів у плануванні [Ознайомленість]
 - визначати шляхи адаптації логіки, що лежить в основі алгоритмів планування, до застосування в інших областях, таких як дискове введення-виведення, мережне планування, планування проектів, а також проблем поза межами інформатики [Застосування]

ОС/Керування пам'яттю

[Нормативний 2 рівня, 3 год.]

Питання:

- Огляд апаратних засобів фізичної пам'яті та керування пам'яттю
- Робочі набори та заміщення сторінок
- Кешування пам'яті (перехресне посилання: Архітектура та організація/Організація та архітектура системи пам'яті)

Результати навчання:

- описувати ієрархію пам'яті комп'ютера та тенденції щодо вартості та продуктивності для різних видів пам'яті [Ознайомленість]
- підсумовувати принципи, що лежать в основі віртуальної пам'яті з точки зору кешування та підкачування [Ознайомленість]
- аналізувати тенденції щодо швидкості доступу до різних видів пам'яті та їх обсягу (основна пам'ять, кеш-пам'ять, зовнішня пам'ять) [Володіння]
- описувати переваги різних шляхів виділення пам'яті для задач [Володіння]
- обґрунтувати потребу використання кеш-пам'яті (продуктивність і наближення, різні виміри проблеми ускладнення абстракцій ізоляції та віртуальної пам'яті через введення кешу) [Ознайомленість]

ОС/Безпека і захист

[Нормативний 2 рівня, 2 год.]

Питання:

- Огляд питань системної безпеки
- Політики безпеки та розмежування доступу

- Методи та пристрої реалізації безпеки системи
- Захист, керування доступом та аутентифікація
- Резервне копіювання

Результати навчання:

- обґрунтовувати потребу захисту і безпеки в операційній системі (перехресне посилання: Інформаційна безпека та захист/Архітектура системи безпеки та системне адміністрування/Дослідження безпеки різних операційних систем) [Володіння]
- підсумовувати особливості та обмеження можливостей операційної системи для захисту та безпеки (перехресне посилання: Інформаційна безпека та захист/Архітектура системи безпеки та системне адміністрування) [Ознайомленість]
- описувати доступні в операційній системі механізми контролю доступу до ресурсів (перехресне посилання: Інформаційна безпека та захист/Архітектура системи безпеки та системне адміністрування/Контроль доступу/Налаштування безпечної роботи систем) [Ознайомленість]
- виконувати прості адміністративні задачі відповідно до політики безпеки, зокрема створення облікових записів, встановлення дозволів, застосування патчів, призначення регулярного резервного копіювання (перехресне посилання: Інформаційна безпека та захист/Архітектура системи безпеки та системне адміністрування) [Застосування]

ОС/Віртуальні машини

[Вибірковий]

Питання:

- Типи віртуалізації (включаючи апаратну/програмну віртуалізацію, віртуалізацію рівня ОС, віртуалізацію серверів, віртуальні сервіси, мережні технології віртуалізації)
- Підкачування та віртуальна пам'ять
- Віртуальні файлові системи
- Гіпервізори
- Портативна віртуалізація; емуляція та ізоляція
- Вартість віртуалізації

Результати навчання:

- пояснювати поняття віртуальної пам'яті та особливості її реалізації на апаратному та програмному рівнях [Ознайомленість]
- розрізняти емуляцію та ізоляцію [Ознайомленість]
- аналізувати тенденції у галузі віртуалізації [Володіння]
- обговорювати гіпервізори та потребу в них з огляду на різні типи гіпервізорів [Застосування]

ОС/Керування пристроями

[Вибірковий]

Питання:

- Характеристика пристроїв послідовного та паралельного доступу
- Абстракції пристроїв та їх особливості
- Стратегії буферизації
- Прямий доступ до пам'яті
- Відновлення після збоїв

Результати навчання:

- пояснювати ключові відмінності між послідовними та паралельними пристроями та визначати умови, за яких використовуються кожен з них [Ознайомленість]
- визначати взаємозв'язки між апаратними та віртуальними пристроями, підтримуваними операційною системою [Застосування]
- пояснювати суть буферизації та описувати стратегії її реалізації [Ознайомленість]
- розрізняти механізми, що використовуються для взаємодії між різними пристроями (включаючи мобільні пристрої, мережні та мультимедійні пристрої) і комп'ютером, та пояснювати особливості застосування цих механізмів під час проектування операційних систем [Застосування]
- описувати переваги та недоліки прямого доступу до пам'яті та обговорювати обставини, за яких його використання виправдане [Застосування]
- визначати вимоги для уможливлення відновлення після збою [Ознайомленість]
- реалізовувати простий драйвер пристрою для ряду можливих пристроїв [Застосування]

ОС/Файлові системи

[Вибірковий]

Питання:

- Файли: дані, метадані, операції, організація, буферизація, послідовний та довільний доступ
- Каталоги: вміст та структура
- Файлові системи: дискові розділи, монтування/розмонтування, віртуальні файлові системи
- Стандартні техніки реалізації
- Відображення файлів у пам'ять
- Спеціальні файлові системи
- Іменування, пошук, доступ, резервне копіювання
- Журналювання та журнальовані файлові системи

Результати навчання:

- описувати можливі варіанти реалізації файлових систем [Ознайомленість]
- порівнювати різні підходи до організації файлів з урахуванням переваг та недоліків кожного з них [Застосування]
- пояснювати, як розвиток апаратних засобів веде до змін пріоритетів у проектуванні файлових систем та керуванні ними [Ознайомленість]
- здійснювати огляд технологій журналювання і пояснювати, як журнальовані файлові системи підвищують відмовостійкість [Ознайомленість]

ОС/Системи реального часу та вбудовані системи*[Вибірковий]***Питання:**

- Планування процесів та задач
- Вимоги до керування основною та дисковою пам'яттю у середовищі реального часу
- Помилки, ризики та відновлення
- Характерні проблеми систем реального часу

Результати навчання:

- пояснювати, що робить систему системою реального часу [Ознайомленість]
- пояснювати причини виникнення затримок у системах реального часу та характеризувати ці затримки [Ознайомленість]
- виокремлювати спеціальні поняття, пов'язані з системами реального часу, включаючи ризик, і пояснювати суть цих понять [Ознайомленість]

ОС/Відмовостійкість*[Вибірковий]***Питання:**

- Фундаментальні поняття: надійні та доступні системи (перехресне посилення: Основи систем/Надійність засобами надмірності)
- Просторова та часова надмірність (перехресне посилення: Основи систем/Надійність засобами надмірності)
- Методи реалізації відмовостійкості
- Приклади механізмів ОС для виявлення, відновлення, перезавантаження з метод реалізації відмовостійкості, використання цих технологій для сервісів ОС

Результати навчання:

- пояснювати значення термінів «відмовостійкість», «надійність» і «доступність» [Ознайомленість]
- окреслювати коло методів реалізації відмовостійкості в операційній системі [Ознайомленість]
- пояснювати, яким чином операційна система може продовжувати функціонування після збою [Ознайомленість]

ОС/Оцінювання продуктивності системи*[Вибірковий]***Питання:**

- Чому продуктивність системи потребує оцінювання (перехресне посилання: Основи систем/Продуктивність/Показники продуктивності)
- Що має вимірюватися (перехресне посилання: Основи систем/Продуктивність/Показники продуктивності)
- Політики продуктивності системи, зокрема кешування, підкачування, планування, керування пам'яттю, безпека
- Моделі вимірювання: детермінована, аналітична, імітаційна, орієнтована на конкретну реалізацію
- Як збирати результати вимірювання (механізми профілювання та відслідковування)

Результати навчання:

- описувати показники вимірювання продуктивності, що використовуються для визначення продуктивності системи [Ознайомленість]
- пояснювати суть основних моделей оцінювання продуктивності систем [Ознайомленість]

Додаток Ж.
ПОРІВНЯННЯ КУРСІВ З ОС У ЗВО СВІТУ

Таблиця Ж.1

Фактичні дані порівняльного аналізу курсів з ОС у ЗВО світу

№ п/п	Де й коли викладався курс	Назва курсу, автори/автори	Які ОС розглядаються	Організація практичної частини курсу	Місце віртуалізації у курсі
1.	Аерокосмічний університет Ембрі-Ріддла (Embry-Riddle Aeronautical University, ERAU), США	«Операційні системи» (CS 420: Operating Systems), Нік Бріксіус (Nick Brixius) [213, с. 330-331]	Windows та UNIX/Linux	Домашні роботи, включаючи проекти з програмування (Java)	Лекції. Розглядається тема «Віртуалізація» (підкачування і віртуальна пам'ять, віртуальні файлові системи, портативна віртуалізація, гіпервізори, методи реалізації віртуальної пам'яті та віртуальних машин). Лабораторні. Немає відомостей про застосування віртуалізації у досліджуваному нами розумінні. Проекти з програмування передбачають використання Java-машини (Java Virtual Machine).
2.	Гарвардський університет (Harvard University), США	«Операційні системи» (CS 161: Operating Systems), Джеймс Мікенс, Марго Селцер (James Mickens, Margo Seltzer) [216]	OS/161 (навчальна ОС)	Практичні завдання (assignments) та підготовчі вправи. Підготовчі вправи (preliminary exercise) передбачають аналіз запропонованих фрагментів коду з відповідями на запитання про цей код, компіляцію та запуск ядра, а також освоєння роботи з дебаггером	Лекції. Розглядається тема «Віртуалізація» (базова ідея, відмінності між інтерфейсом монітора віртуальних машин та інтерфейсом операційної системи, сфери застосування віртуалізації, підходи до реалізації віртуалізації та їх порівняння). Лабораторні. Раніше використовувався емулятор System/161 (гостьова ОС – Nachos) [231], [209]. Зараз застосовується CS50 Appliance (заздалегідь налаштований образ віртуальної машини на базі Ubuntu Linux) у поєднанні з VMware Fusion (якщо основна ОС – Mac OS X), VMware Workstation (якщо основна ОС – Windows чи Linux, 64-розрядні версії) або VirtualBox (у решті випадків).

№ п/п	Де й коли викладався курс	Назва курсу, автори/автори	Які ОС розглядаються	Організація практичної частини курсу	Місце віртуалізації у курсі
3.	Гельсінський університет (University of Helsinki), Фінляндія, 2015	«Операційні системи» (582219: Operating Systems), Тіму Керола (Teemu Kerola) [238], [213, с. 334-335]	Linux, Windows	Домашні роботи	Лекції. Розглядається тема «Віртуалізація» (гіпервізори, монітори віртуальних машин, реалізація віртуальних машин, віртуальна пам'ять, віртуальні файлові системи). Лабораторні. Немає відомостей про застосування віртуалізації у досліджуваному нами розумінні.
4.	Єльський університет (Yale University), США, осінь 2015	«Проектування і розробка операційних систем» (CS 422/522: Design and Implementation of Operating Systems), Чжун Шао (Zhong Shao) [215]	Ядро mCertiKOS	Практичні завдання (assignments)	Лекції. Тема «Віртуалізація» безпосередньо не розглядається. Лабораторні. Використовується емулятор QEMU та гіпервізор mCertiKOS (на базі ядра CertiKOS, Certified Kernel for Secure Cloud Computing [228]).
5.	Кембриджський університет (University of Cambridge), Великобританія	«Випікання пирога – Розробка операційних систем» (Baking Pi – Operating Systems Development), Алекс Чедвік (Alex Chadwick) [202]	Пишеться власна	Уроки (крок за кроком створюється власна проста ОС для одноплатного комп'ютера Raspberry Pie)	Лекції. Тема «Віртуалізація» безпосередньо не розглядається. Лабораторні. Окреме віртуалізоване середовище для кожного студента не використовується – його студентіві заміняє Raspberry Pie
6.	Кембриджський університет (University of Cambridge), Великобританія, 20015-16 н.р.	«Операційні системи» (Operating Systems), Ричард Морт'є (Richard Mortier) [261]	Unix, Windows NT	Передусім теоретичний курс. Є основою для наступних курсів, у тому числі для «Поглибленого курсу з операційних систем» (Advanced Operating Systems, L41), який доступний студентам зі ступенем бакалавра і має більш практичний характер (розглядається ОС FreeBSD) [196]	Лекції. Тема «Віртуалізація» безпосередньо не розглядається. Лабораторні. Немає відомостей про застосування віртуалізації у досліджуваному нами розумінні.

№ п/п	Де й коли викладався курс	Назва курсу, автори/автори	Які ОС розглядаються	Організація практичної частини курсу	Місце віртуалізації у курсі
7.	Коледж Уільямса (Williams College), США	«Операційні системи» (CSCI432: Operating Systems), Джині Олбрет (Jeannie Albrecht) [219]	Unix, Windows, Mach, Mac OS	Проекти, письмові домашні завдання, письмові дослідницькі роботи	Лекції. Розглядається тема «Віртуалізація» (типи віртуалізації, будова різних гіпервізорів, тенденції). Лабораторні. Немає відомостей про застосування віртуалізації у досліджуваному нами розумінні. Використовуються комп'ютерні класи з ОС UNIX.
8.	Колумбійський університет (Columbia University), США, осінь 2016	«Операційні системи» (COMS W4118 Operating Systems), Джейсон Ньє (Jason Nieh) [259]	Android, Linux	Домашні роботи (здача відбувається через систему контролю версій Git)	Лекції. У версії курсу 2013 року розглядається тема «Віртуалізація» Лабораторні. Домашні роботи виконуються на наданій віртуальній машині, яка запускається на гіпервізорі VMware Workstation (якщо основна ОС – Windows або Linux) або VMware Fusion (якщо основна ОС – Mac OS X). Ліцензії для VMware Workstation/Fusion надаються безкоштовно через студентські облікові записи (CS accounts). ПЗ встановлюється на студентські ноутбуки. Це перший курс, під час викладання якого було залучено технології віртуалізації як засіб навчання інформатики, зокрема ОС (з 2000 року) [253].
9.	Массачусетський технологічний інститут (Massachusetts Institute of Technology, MIT), США, осінь 2014	«Інженерія операційних систем» (6.828: Operating Systems Engineering), Франц Кашук (Frans Kaashoek) [262; 258]	xv6 (навчальна)	Лабораторні роботи, а також домашні завдання (до кожної лекції), огляди чужого коду, проблемні завдання, опитування, підсумковий проект	Лекції. Розглядається тема «Віртуальні машини» (лекція 16). Лабораторні. Використовується емулятор QEMU, а також, за потреби, Cygwin (для забезпечення функціоналу, близького до Linux, у Windows-системах).
10.	Принстонський університет (Princeton University), США, осінь 2015	«Операційні системи» (CS318: Operating Systems), Джесвіндер П. Сінг (Jaswinder P. Singh) [212]	Linux	Проекти з програмування	Лекції. Розглядається тема «Монітори віртуальних машин». Лабораторні. Застосовуються емулятори Bochs та QEMU в окремо виділеній комп'ютерній лабораторії. Можливий віддалений доступ до комп'ютерів лабораторії. Як альтернативний варіант передбачене використання VirtualBox і готових образів Linux зі встановленим емулятором Bochs (для користувачів Windows і Mac OS X) [194; 245]

№ п/п	Де й коли викладався курс	Назва курсу, автори/автори	Які ОС розглядаються	Організація практичної частини курсу	Місце віртуалізації у курсі
11.	Рейк'явіцький університет (Reykjavik University), Ісландія	«Операційні системи» (RU STY1 Operating Systems), Імір Вігфюссон (Ymir Vigfusson) [213, с. 336-338]	CentOS 5 Linux	Відомості відсутні	Лекції. Тема «Віртуалізація» безпосередньо не розглядається. Лабораторні. Для роботи студентам надається термінальний доступ до серверу з CentOS Linux, однак деталей не наводиться.
12.	Система масових відкритих онлайн курсів Udacity (Udacity MOOC) / Технологічний інститут Джорджії (Georgia Institute of Technology, Georgia Tech), США	«Вступ до операційних систем» (Introduction to Operating Systems), Ада Гавриловська, Джерод Паркс (Ada Gavrilovska, Jarrod Parkes) [233; 234], базується на курсі CS8803: Introduction to Operating Systems, що викладається у Технологічному інституті Джорджії (Georgia Tech) [218]	Ubuntu Linux	Проекти з програмування, вправи (онлайн-опитування)	Лекції. Тема «Віртуалізація» безпосередньо не розглядається. Лабораторні. Готові віртуальні машини з Ubuntu Linux на базі гіпервізора VirtualBox.
13.	Стенфордський університет (Stanford University), США, зима 2016	«Операційні системи» (CS140: Operating Systems, Джон Аустерхаут (John Ousterhout) [214]	Pintos (навчальна ОС)	Проекти з програмування	Лекції. Розглядається тема «Віртуальні машини» (лекції 26-27). Лабораторні. Використовуються емулятори Bochs та QEMU.
14.	Університет Арканзасу, Літл Рок (University of Arkansas at Little Rock, UALR), США	«Операційні системи» (CPSC 3380 Operating Systems) [213, с. 332-333], Пейї Тан (Peiyi Tang)	Nachos (навчальна ОС)	Відомості відсутні	Лекції. Розглядається тема «Віртуалізація» (обґрунтування потреби у захисті та передбачуваній продуктивності, рівні перескєрування, методи реалізації віртуальної пам'яті та віртуальних машин, підкачування та віртуальна пам'ять, віртуальні файлові системи, портативна віртуалізація, гіпервізори). Лабораторні. Немає відомостей про застосування віртуалізації у досліджуваному нами розумінні.

№ п/п	Де й коли викладався курс	Назва курсу, автори/автори	Які ОС розглядаються	Організація практичної частини курсу	Місце віртуалізації у курсі
15.	Університет Каліфорнії, Берклі (University of California, Berkeley), США, весна 2016	«Операційні системи та системне програмування» (CS162: Operating Systems and System Programming), Ентоні Д. Джозеф, Йон Стоїка (Anthony D. Joseph, Ion Stoica) [217]	Nachos (навчальна ОС)	Домашні роботи, проекти з програмування, тестові завдання (для захисту кожного проекту)	Лекції. Тема «Віртуалізація» безпосередньо не розглядається. Лабораторні. Використовуються віртуальні машини (готові, надаються) на базі віртуалізаційного ПЗ Vagrant.
16.	Університет Карнегі-Меллона (Carnegie Mellon University, CMU), США, весна 2016	«Проектування та розробка операційних систем» (15-410, Operating System Design and Implementation), Дейв Екхардт (Dave Eckhardt) [257]	Linux	Домашні роботи, проекти з програмування	Лекції. Розглядається тема «Віртуалізація» (обґрунтування потреби у віртуалізації, емуляція, динамічна трансляція, типи інструкцій, віртуалізація платформи x86, паравіртуалізація). Лабораторні. Використовується емулятор Simics.
17.	Федеральна вища школа Цюриха (Swiss Federal Institute of Technology, ETH Zurich), Швейцарія	«Операційні системи та комп'ютерні мережі» (Operating Systems and Networks), Дональд Косман, Торстен Гоефлер (Donald Kossmann, Torsten Hoeffler) [260]	Barrelfish OS (експериментальна ОС для багатоядерних ЦП)	Лабораторні роботи і відповіді на запитання	Лекції. Розглядається тема «Монітори віртуальних машин». Лабораторні. Застосовується емулятор QEMU.

Додаток 3.**ПРИКЛАД НЕФОРМАЛЬНОЇ ДОКУМЕНТАЦІЇ ДЛЯ ФІКСУВАННЯ
ВИКЛАДАЧЕМ ПРОЦЕСУ ПРОЕКТУВАННЯ МЕТОДИКИ****Викладацька документація: цільовий блок****Соціальне замовлення***Вимоги роботодавців (+ суміжні галузі)**Тенденції у предметній галузі та на ринку праці***Компетентності та уміння з галузевого стандарту****Вдосконалення методики**

Викладацька документація: лекційний блок

№ п/п	Форма навчання	Тема лекції	Кількість годин	Виправлення			
				Структурні зауваження	Зміни до матеріалів лекції	Організаційно-технічні зауваження	Інші позначки
	(Лекція- бесіда, лекція- презентація тощо)			(Скоротити / Розширити / Оновити / Суттєво переробити / Переробити повністю тощо)	(Що додати / уточнити / оновити / замінити / прибрати / структурувати тощо)	(Необхідне матеріальне забезпечення та технічні засоби / Роздаткові матеріали / Дії у разі непередбачуваних технічних збоїв і т.д.)	

Викладацька документація: лабораторний блок

№ п/п	Вид роботи	Тема роботи	Кількість годин	Виправлення			
				Структурні зауваження	Зміни до інструктивно-методичних матеріалів	Організаційно-технічні зауваження	Інші позначки
	<i>(Лабораторна робота / Самостійна робота тощо)</i>			<i>(Розділити / Об'єднати / Оновити / Суттєво переробити / Переробити повністю тощо)</i>	<i>(Що додати / уточнити / виправити / оновити / прибрати тощо)</i>	<i>(Необхідне матеріальне забезпечення та технічні засоби / Роздаткові матеріали / Дії у разі непередбачуваних технічних збоїв і т.д.)</i>	

Додаток И.

ПОРІВНЯННЯ ВІРТУАЛІЗАЦІЙНОГО ПЗ У НАВЧАННІ UNIX-ПОДІБНИХ ОС БАКАЛАВРІВ ІНФОРМАТИКИ

Таблиця И.1

Порівняльний аналіз програмних засобів віртуалізації у навчанні Unix-подібних ОС бакалаврів інформатики:
фактичні відомості

Назва ПЗ	Критерії та показники	Відповідність технології віртуалізації потребам курсу (1)	Універсальність (2)		Ліцензійна чистота (3)	Функціональність інтерфейсу (4)				Популярність та підтримка (5)		
		Показники вибору технології віртуалізації за напрямом / за методом (1.1) / (1.2)	Універсальність щодо основних ОС / гостей ОС (2.1) / (2.2)	Універсальність щодо технологій апаратної віртуалізації (2.3)	Вид ліцензії (3.1)	Наявність графічного інтерфейсу (4.1)	Керованість із командного рядка (4.2)	Наявність українського інтерфейсу (4.3)	Можливість віддаленого доступу (4.4)	СПІ (5.1)	Засоби підтримки користувачів (5.2)	Дата випуску останнього стабільного релізу (станом на січень 2015) (5.3)
Oracle VirtualBox [265]	гіпервізор ІІ типу, динамічна трансляція	Windows, Mac OS X, Linux, Solaris / Windows, FreeBSD, OS/2, QNX, Linux, Mac OS X, Solaris, Syllable та ін.	апаратна віртуалізація підтримується, але не є обов'язковою	вільно поширюване ПЗ (GNU GPL v.2), деякі компоненти – PUEL (умовно безкоштовна, в тому числі для всіх навчальних закладів)	так	так	так	так, VBoxWeb, phpVirtualBox та ін.	51	користувальська та технічна документація, відкриті коди, форум, списки розсилки, IRC та ін.	листопад 2014	
VMware Player [290]	гіпервізор ІІ типу, динамічна трансляція	Windows, Linux / Windows, FreeBSD, OS/2, Mac OS X, Netware, Toshiba, Linux та ін.	апаратна віртуалізація підтримується, але не є обов'язковою	умовно безкоштовне з доступом до кодів – для некомерційного користування та використання у навчальних закладах VMware Academic Program, інакше платне	так	так	ні	ні	12	користувальська та технічна документація, база знань, відкриті коди, форум та ін.	грудень 2014	

	(1)	(2)		(3)	(4)				(5)		
	(1.1) / (1.2)	(2.1) / (2.2)	(2.3)	(3.1)	(4.1)	(4.2)	(4.3)	(4.4)	(5.1)	(5.2)	(5.3)
Kernel Based Virtual Machine (KVM) [239; 240]	гіпервізор II типу, динамічна трансляція + паравіртуалізація	Linux / Windows, Linux, OpenBSD, FreeBSD, DragonflyBSD, Solaris, OpenSolaris, QNX та ін.	апаратна віртуалізація підтримується і є обов'язковою	вільно поширюване ПЗ (GNU GPL та LGPL)	так, ПЗ сторонніх розробників	так	ні	так, Witsbits, Eucalyptus, Proxmox та ін.	1	користувацька та технічна документація, відкриті коди, списки розсилки, IRC та ін.	вересень 2012
Proxmox VE [271] + KVM + Open VZ (станом на січень 2015 року, згодом Open VZ замінено на LXC [288; 246])	гіпервізор II типу + віртуальні контейнери + динамічна трансляція + паравіртуалізація	Linux / Windows, Linux, OpenBSD, FreeBSD, DragonflyBSD, Solaris, OpenSolaris, QNX та ін.	апаратна віртуалізація підтримується і є обов'язковою для віртуальних машин на базі KVM та необов'язковою для решти машин	вільно поширюване ПЗ (GNU AGPL v.3)	так	так	ні	так, інтерфейсом для віддаленого доступу є саме ПЗ	1	користувацька та технічна документація, відеоуроки, відкриті коди, списки розсилки, а також додаткова платна підтримка	вересень 2014
OpenVZ [256]	віртуальні контейнери	– / Linux	апаратна віртуалізація не підтримується	вільно поширюване ПЗ (GNU GPL v.2)	так, ПЗ сторонніх розробників	так	ні	так, OpenVZ Web Panel, Proxmox VE та ін.	2	користувацька та технічна документація, відкриті коди, форуми, список розсилки та ін.	січень 2015
FreeBSD jail [235]	віртуальні контейнери	– / FreeBSD	апаратна віртуалізація не підтримується	вільно поширюване ПЗ (FreeBSD License)	так, ПЗ сторонніх розробників	так	ні	так, CBSD та ін.	0	користувацька та технічна документація, відкриті коди	листопад 2014
Linux-Vserver [291]	віртуальні контейнери	– / Linux	апаратна віртуалізація не підтримується	вільно поширюване ПЗ (GNU GPL v.2)	так, ПЗ сторонніх розробників	так	ні	так, OpenVCP та ін.	0	користувацька та технічна документація, відкриті коди, списки розсилки, IRC	квітень 2007 (експериментальні версії: січень 2015)

	(1)	(2)		(3)	(4)				(5)		
	(1.1) / (1.2)	(2.1) / (2.2)	(2.3)	(3.1)	(4.1)	(4.2)	(4.3)	(4.4)	(5.1)	(5.2)	(5.3)
Oracle VM Server for x86 [264]	гібридний гіпервізор; паравіртуалізація (для ОС із відкритим кодом) + апаратна віртуалізація (для ОС із закритим кодом)	– / Linux (офіційно: Oracle Linux, Red Hat Enterprise Linux, Oracle Solaris, Windows Server, Windows 7/8 та ін.)	апаратна віртуалізація підтримується і є обов'язковою для ОС із закритим кодом	Oracle VM Server: вільно поширюване ПЗ (GNU GPL) Oracle VM Manager: безкоштовне (private source, freely re-distributable, free use)	так	так	ні	так, інтерфейс доступу включено до ПЗ	2	оглядова користувацька та технічна документація; підтримка переважно є платною	липень 2014
XenServer [295]	гібридний гіпервізор; паравіртуалізація (для ОС із відкритим кодом) + апаратна віртуалізація (для ОС із закритим кодом)	– / Windows, Linux, Minix, Plan 9, NetBSD, FreeBSD, OpenSolaris, NetWare та ін.	апаратна віртуалізація підтримується і є обов'язковою для ОС із закритим кодом	вільно поширюване ПЗ (GNU GPL v.2)	так	так	ні	так, HyperVM, ConVirt, OpenNebula, YaST, Xen Orchestra та ін.	21	користувацька та технічна документація, відкриті коди, списки розсилки, IRC	січень 2015

Додаток К.

ЗМІСТ ЛЕКЦІЙ ТА ЛАБОРАТОРНИХ РОБІТ З НАВЧАЛЬНОЇ ДИСЦИПЛІНИ «ОПЕРАЦІЙНІ СИСТЕМИ ТА СИСТЕМНЕ ПРОГРАМУВАННЯ»

Таблиця. К.1

Зміст лекцій з навчальної дисципліни «Операційні системи та системне програмування»

Тема, питання	К-ть годин	Результати навчання	Ключові поняття	Змістові модулі з CS2013
Тема 1. Початкові відомості про операційні системи ОС як розширена машина та менеджер ресурсів. Історія розвитку ОС. Структура ОС. Режим ядра та режим користувача. Ядро. Класифікація ОС за архітектурою. Класифікація ОС за сферою застосування.	2	Уміти пояснювати роль ОС у взаємодії користувача з комп'ютером та в керуванні системними ресурсами. Уміти пояснювати роль абстракцій у роботі ОС. Орієнтуватися в історії розвитку ОС і на її основі ілюструвати взаємозв'язок між розвитком апаратного та програмного забезпечення. Знати, за яких умов ОС вважається мережною. Знати відмінності розподілених ОС від традиційних ОС. Уміти порівнювати режим ядра та режим користувача ОС. Орієнтуватися в архітектурах ОС (монолітні, багаторівневі, модульні, мікроядра), перевагах та недоліках цих архітектур. Уміти характеризувати ОС залежно від сфери їх застосування (від ОС мейнфреймів до ОС смарт-карт). Уміти аналізувати спадковість тенденцій у проектуванні ОС.	Операційна система; системні ресурси; ядро; системний виклик; багатозадачність	ОС/Огляд ОС, ОС/Основні принципи роботи ОС
Тема 2. Процеси та потоки Процес як програма під час виконання. Ієрархія процесів. Потік. Процес як абстракція для групування ресурсів. Моделі процесів та потоків. Сигнали. Обробка переривань. Системні виклики POSIX та виклики Win32 API. Структури даних для зберігання відомостей про процеси та потоки. Інтерактивні та фонові процеси. Реалізація процесів та потоків у Linux та у Windows	3	Уміти характеризувати роль ієрархії процесів у Unix-подібних ОС та у Windows. Уміти характеризувати моделі ОС, залежно від кількості у ній процесів та потоків. Уміти обґрунтовувати потребу у багатозадачності у сучасних ОС. Уміти пояснювати відмінність псевдопаралелізму від реального паралелізму у багатопроекторних системах. Уміти характеризувати типові стани процесів, а також умови та особливості переходу процесу між цими станами. Уміти порівнювати системні виклики стандарту POSIX та виклики інтерфейсу прикладного програмування Win32 API. Уміти описувати призначення основних типових структур даних для зберігання відомостей про процеси та потоки. Уміти порівнювати особливості реалізації процесів та потоків у Linux та Windows.	Процес; адресний простір; потік; стан; перемикання контекстів; переривання; сигнал; багатозадачність; POSIX; API; блок керування процесом (PCB); блок керування потоком (TCB); ідентифікатор процесу (PID)	ОС/Основні принципи роботи ОС, ОС/Паралелізм, ОС/Планування та диспетчеризація

Тема, питання	К-ть годин	Результати навчання	Ключові поняття	Змістові модулі з CS2013
Тема 3. Пам'ять Ієрархія пам'яті комп'ютера. Фізичні та віртуальні адреси. Підходи до організації пам'яті (підхід базового і межового регістрів, сегментна, сторінкова, сторінково-сегментна організація пам'яті, багаторівневі таблиці сторінок, кеш трансляції). Підкачування. Алгоритми заміщення сторінок	2	Уміти описувати ієрархію пам'яті комп'ютера, тенденції щодо вартості та обсягів різних видів пам'яті, а також швидкості доступу до них. Уміти характеризувати основні підходи до організації пам'яті), переваги та недоліки кожного з них. Уміти пояснювати суть явища фрагментації, відмінності зовнішньої й внутрішньої фрагментації. Уміти обґрунтовувати потребу в механізмах кешування, підкачування. Розуміти суть основних алгоритмів заміщення сторінок, їх переваги та недоліки.	Фізична та віртуальна; пристрій керування пам'яттю; сегмент; сторінка; кеш трансляції; фрагментація; підкачування; заміщення сторінок	ОС/Керуван-ня пам'яттю, ОС/Віртуаль-ні машини, Архітектура та організа-ція та архітектура системи пам'яті)
Тема 4. Файлові системи Файл. Особливості іменування файлів. Файлові атрибути. Каталог як група файлів. Каталог як спеціальний файл. Абсолютне та відносне ім'я файлу. Дискові розділи. Огляд конкретних файлових систем (FAT, NTFS, UFS, ext). Сумісність файлових систем	2	Знати типові файлові атрибути. Уміти порівнювати правила іменування файлів у файлових системах для Unix-подібних ОС та файлових системах для Windows. Уміти описувати призначення дискових розділів, їх роль у завантаженні ОС. Уміти характеризувати основи будови та функціонування конкретних файлових систем (FAT, NTFS, UFS, ext). Уміти обґрунтовувати потребу та пояснювати суть технології журналювання. Уміти пояснювати роль віртуальних файлових систем у реалізації сумісності файлових систем.	Файл; атрибут; каталог; абсолютне та відносне ім'я; файлова система; журналювання; віртуальна файлова система.	ОС/Файлові системи, ОС/Безпека і захист, ОС/Віртуаль-ні машини
Тема 5. Безпека Огляд питань системної безпеки. Ідентифікація, аутентифікація, авторизація. Типи, форми та деякі варіанти реалізації аутентифікації. Аутентифікація у Unix-подібних системах та у Windows. Списки керування доступом та переліки можливостей. Принцип мінімальних повноважень. Авторизація у Unix-подібних системах та у Windows. Аудит. Політика аудиту. Журнал аудиту. Реалізація аудиту у Unix-подібних ОС та у Windows	3	Уміти визначати потенційні загрози для ОС. Уміти пояснювати зв'язки та відмінності між процедурами ідентифікації, аутентифікації, авторизації. Уміти характеризувати типи аутентифікації (локальна, мережна), її форми (із використанням паролю, фізичного об'єкту, біометричних параметрів), описувати конкретні варіанти аутентифікації (на основі односторонніх функцій, одноразових паролів тощо). Уміти описувати основні компоненти системи безпеки у Linux та у Windows. Уміти пояснювати суть принципу мінімальних повноважень. Уміти порівнювати авторизацію на основі списків керування доступом та авторизацію за допомогою переліків можливостей. Уміти характеризувати основні особливості авторизації у Linux та Windows. Уміти обґрунтовувати необхідність аудиту в ОС. Уміти описувати основи організації аудиту у Linux та Windows.	Ідентифікація; аутентифікація; авторизація; списки керування доступом; переліки можливостей; обліковий запис; група; аудит; політика аудиту; журнал аудиту	ОС/Безпека і захист

Тема, питання	К-ть годин	Результати навчання	Ключові поняття	Змістові модулі з CS2013
Тема 6. Планування Планування та вимоги до нього. Планування без витіснення та планування з витісненням. Алгоритми планування у системах пакетної обробки, інтерактивних системах, системах реального часу. Особливості планування у Linux та Windows	2	Уміти характеризувати основні вимоги до планування. Уміти пояснювати залежність вибору об'єкта планування в ОС від моделі реалізації потоків. Уміти пояснювати відмінність між процесами, обмеженими швидкістю обчислень, та процесами, обмеженими швидкістю роботи пристроїв введення-виведення. Уміти порівнювати планування без витіснення та з витісненням. Уміти характеризувати підходи до планування залежно від сфери застосування (системи пакетної обробки, інтерактивні системи, системи реального часу) та описувати суть роботи основних алгоритмів для кожної із цих сфер. Уміти описувати основи планування у Linux та Windows (об'єкт планування, алгоритми планування, алгоритми, система пріоритетів).	Планувальник; планування без витіснення; планування з витісненням; пріоритет; квант	ОС/Планування та диспетчеризація
Тема 7. Міжпроцесова та міжпоточкова взаємодія Проблема міжпроцесової взаємодії. Задача синхронізації. Синхронізаційні механізми (семафори, м'ютекси, умовні змінні, монітори, блокування читання-запису, бар'єри тощо). Передача даних між процесами (сигнали, спільно використовувана пам'ять, повідомлення, відображувана пам'ять)	2	Уміти обґрунтовувати потребу у міжпроцесовій та міжпоточковій взаємодії. Уміти наводити приклади потенційних проблем, які можуть виникати під час виконання паралельних операцій кількома різними задачами. Уміти характеризувати основні синхронізаційні примітиви (семафори, м'ютекси, умовні змінні, монітори, бар'єри тощо). Уміти характеризувати основні механізми передачі даних між процесами (сигнали, спільно використовувана пам'ять, повідомлення, відображувана пам'ять).	Спільно використовуваний ресурс; критична область; змагання (перегони); семафор; м'ютекс; умовна змінна; монітор; бар'єр; відображувана пам'ять	ОС/Паралелізм
Тема 8. Віртуалізація* Поняття віртуалізації. Напрями та методи віртуалізації. Віртуалізація серверів. Віртуальні машини та віртуальні контейнери. Віртуальний хостинг. Хмарні технології	2	Орієнтуватися у різних тлумаченнях терміну «віртуалізація». Уміти пояснювати місце віртуалізації в операційній системі. Уміти характеризувати основні напрями та методи віртуалізації. Уміти аналізувати доцільність вибору засобу віртуалізації залежно від сфери застосування, наявних ресурсів та вимог. Уміти пояснювати зв'язок між технологіями віртуалізації та хмарними технологіями.	Віртуалізація; віртуальна машина; гіпервізор; віртуальний контейнер; віртуальний сервер; хмарний сервіс	ОС/Віртуальні машини
Загальна кількість годин:	18			

Таблиця К.2

Зміст лабораторних робіт з навчальної дисципліни «Операційні системи та системне програмування»

Лабораторна робота	Потребує теоретичних відомостей з тем	Засоби віртуалізації, інше ПЗ	Результати навчання
Лабораторна робота №1. Встановлення Linux	Початкові відомості про операційні системи	VirtualBox без віртуалізації	Знати історію виникнення ОС Linux. Уміти розрізняти поняття ядра Linux, дистрибутиву Linux та випуску Linux. Знати історію створення дистрибутиву Ubuntu, його характерні особливості та версії. Уміти здійснювати встановлення Linux за допомогою мережного інсталятора.
Лабораторна робота №2. Знайомство з Linux	Початкові відомості про операційні системи	VirtualBox Amazon EC2 без віртуалізації	Орієнтуватися у віконних системах для Linux та окреслювати коло ситуацій, коли потрібний графічний інтерфейс. Уміти пояснювати суть та призначення механізму sudo. Орієнтуватися в основних елементах робочого столу та меню Linux (наприкладі Xubuntu Linux). Уміти здійснювати навігацію та найпростіші операції з файлами у графічному файловому менеджері (наприкладі Thunar). Уміти аналізувати вміст запрошення командного рядка. Уміти користуватися буфером хронології команд. Уміти виводити вміст каталогу, у тому числі відформатований та відфільтрований, за допомогою команди ls. Уміти виводити вміст файлу за допомогою команди cat. Володіти прийомами посторінкового виведення (команди more, less). Уміти здійснювати пошук у виводі команд за ключовим словом. Уміти здійснювати просту конвеєризацию двох команд та перескерування виводу команди у файл. Уміти одержувати та використовувати довідку у форматі man (за допомогою команди та на веб-ресурсах). Уміти одержувати та використовувати довідку за допомогою команди help. Уміти виконувати команди з правами адміністратора. Уміти завершувати роботу засобами графічного інтерфейсу. Уміти завершувати роботу за допомогою команди shutdown.

Лабораторна робота	Потребує теоретичних відомостей з тем	Засоби віртуалізації, інше ПЗ	Результати навчання
Лабораторна робота №3. Налаштування мережного з'єднання у Linux	Початкові відомості про операційні системи	VirtualBox Amazon EC2 без віртуалізації	Уміти здійснювати основні операції у простому текстовому редакторі з графічним інтерфейсом (наприкладі Mousepad). Уміти здійснювати основні операції у простому текстовому редакторі з інтерфейсом командного рядка (наприкладі nano). Уміти пояснювати призначення конфігураційних файлів та окреслювати коло ситуацій, коли доцільно редагувати їх вручну. Уміти перевіряти роботу мережного з'єднання шляхом пінгування зовнішньої адреси. Уміти інтерпретувати вміст файлу /etc/network/interfaces. Відповідно до синтаксису файлу /etc/network/interfaces, вміти описувати простий інтерфейс Ethernet (методи DHCP та static) та loopback-інтерфейс. Уміти характеризувати призначення віртуального мережного loopback-інтерфейсу. Уміти одержувати та інтерпретувати відомості про мережні інтерфейси за допомогою команди ifconfig. Уміти встановлювати статичну конфігурацію для інтерфейсу Ethernet за допомогою команди ifconfig. Уміти встановлювати динамічну конфігурацію для інтерфейсу Ethernet за допомогою команди dhclient. Уміти одержувати та інтерпретувати відомості про наявні мережні з'єднання та створювати нові засобами графічного інтерфейсу (наприкладі GUI-засобу Мережні з'єднання). Уміти підключатися до мережі та відключитися від неї засобами графічного інтерфейсу. Уміти здійснювати налаштування з'єднання через проксі-сервер шляхом редагування конфігураційних файлів.

Лабораторна робота	Потребує теоретичних відомостей з тем	Засоби віртуалізації, інше ПЗ	Результати навчання
Лабораторна робота №4. Встановлення ПЗ у Linux	Початкові відомості про операційні системи	VirtualBox Amazon EC2 без віртуалізації	Уміти пояснювати роль пакунків та репозиторіїв у встановленні ПЗ у Linux. Уміти описувати особливості організації офіційних репозиторіїв Ubuntu. Орієнтуватися у найпоширеніших форматах пакунків. Уміти порівнювати інструменти для роботи з пакунками у deb-сумісних дистрибутивах, зокрема з точки зору автоматичного врахування залежностей. Уміти встановлювати та вилучати пакунки, а також одержувати відомості про вже встановлені пакунки за допомогою команди dpkg. Уміти аналізувати та редагувати вміст файлу /etc/apt/sources.list Уміти встановлювати та вилучати пакунки, здійснювати пошук пакунків у підключених репозиторіях, автоматичне вилучення непотрібних пакунків та deb-файлів за допомогою утиліти apt. Уміти встановлювати та вилучати пакунки, здійснювати пошук пакунків у репозиторіях за допомогою менеджера пакунків Synaptic. Уміти аналізувати та змінювати налаштування утиліти apt за допомогою GUI-засобу Програмне забезпечення та оновлення.
Лабораторна робота №5. Керування процесами у Linux Спостереження за роботою процесів та потоків у Linux. Керування процесами у Linux GUI-засобами та засобами командного рядка.	Початкові відомості про операційні системи. Процеси та потоки. <i>Планування (реком.)</i>	VirtualBox Amazon EC2 без віртуалізації	Уміти характеризувати стани процесів у Linux. Уміти пояснювати роль рівня пріоритету у Linux. Уміти одержувати та інтерпретувати характеристики процесів за допомогою Диспетчера задач, утиліти top. Уміти запускати, призупиняти, поновлювати та завершувати процеси засобами Диспетчера задач. Уміти одержувати та інтерпретувати відомості про процеси, у тому числі відфільтровані, за допомогою команди ps. Уміти здійснювати звичайне і примусове завершення процесів за допомогою команд kill та killall. Уміти запускати процес у фоновому режимі. Уміти змінювати пріоритет процесу засобами команд nice та renice. Уміти здійснювати завершення та зміну пріоритету процесу засобами утиліти top.

Лабораторна робота	Потребує теоретичних відомостей з тем	Засоби віртуалізації, інше ПЗ	Результати навчання
Лабораторна робота №6. Керування процесами у Windows	Початкові відомості про операційні системи. Процеси та потоки. Планування*	Amazon EC2 без віртуалізації	Уміти характеризувати типи запуску та стани служб у Windows. Уміти одержувати та інтерпретувати характеристики процесів (задач та процесів) за допомогою Диспетчера задач. Уміти завершувати процеси за допомогою Диспетчера задач. Уміти одержувати та інтерпретувати характеристики служб за допомогою оснащення Служби. Уміти зупиняти та запускати служби, змінювати тип служби за допомогою оснащення Служби. Уміти запускати програми з командного рядка. Уміти виводити відомості, у тому числі відформатовані, про процеси за допомогою команди tasklist. Уміти завершувати процеси за допомогою команди taskkill.
Лабораторна робота №7. Програмування процесів та потоків у Windows	Початкові відомості про операційні системи. Процеси та потоки	без віртуалізації, Code::Blocks Amazon EC2, Code::Blocks	Уміти порівнювати системні виклики стандарту POSIX та виклики інтерфейсу Win32 API. Уміти здійснювати пошук довідкової інформації щодо використання Win32 API в MSDN Library. Уміти характеризувати призначення основних типів даних Windows. Уміти пояснювати суть та призначення угорської нотації, характеризувати її місце серед інших підходів до формування ідентифікаторів. Уміти використовувати API-функції CreateProcess(), CloseHandle(), CreateThread(), WaitForSingleObject(), WaitForMultipleObjects(), GetExitCodeProcess().
Лабораторна робота №8. Програмування процесів та потоків у Linux	Початкові відомості про операційні системи. Процеси та потоки	VirtualBox, g++ VirtualBox, g++, Code::Blocks Amazon EC2, g++	Уміти порівнювати системні виклики стандарту POSIX та виклики інтерфейсу Win32 API. Уміти здійснювати пошук довідкової інформації щодо використання бібліотечних функцій стандарту POSIX у документації man. Уміти пояснювати механізм роботи технології fork+exec. Уміти компілювати програмний код засобами компілятора g++ з командного рядка (у тому числі, з підключенням бібліотеки pthread). Уміти запускати на виконання відкомпільований файл програми засобами командного рядка. Уміти використовувати функції fork(), execve(), waitpid(), exit(), pthread_create(), pthread_join().

Лабораторна робота	Потребує теоретичних відомостей з тем	Засоби віртуалізації, інше ПЗ	Результати навчання
Лабораторна робота №9. Моніторинг використання пам'яті та процесорного часу у Linux та Windows	Початкові відомості про ОС. Процеси та потоки. Пам'ять	(для Linux) VirtualBox Amazon EC2 без віртуалізації (для Windows) Amazon EC2 без віртуалізації	Уміти обґрунтовувати потребу в механізмах кешування, підкачування та пояснювати їх суть. Уміти одержувати та аналізувати відомості про використання пам'яті у Linux за допомогою команди free. Уміти одержувати та аналізувати статистику використання віртуальної пам'яті у Linux за допомогою команди vmstat. Уміти визначати, який саме блоковий файл відповідає розділу підкачування у Linux. Уміти одержувати та аналізувати відомості про використання пам'яті та часу ЦП у Windows за допомогою Диспетчера задач (Task Manager). Уміти показувати розташування файлу підкачування у Windows.
Лабораторна робота №10. Робота з файлами у Linux	Початкові відомості про ОС. Процеси та потоки. Пам'ять. Файлові системи	VirtualBox Amazon EC2 без віртуалізації	Уміти здійснювати навігацію та найпростіші операції з файлами у графічному файловому менеджері (на прикладі Thunar). Уміти здійснювати навігацію каталогами за допомогою команди cd. Уміти виводити вміст каталогу, у тому числі відформатований та відфільтрований, за допомогою команди ls. Уміти створювати порожній файл засобами команди touch. Уміти створювати нові каталоги за допомогою команди mkdir. Уміти копіювати, переміщувати, перейменовувати, видаляти файли та каталоги за допомогою команд cp, mv, rm. Уміти пояснювати суть механізму монтування. Уміти здійснювати автоматичне/ручне монтування та розмонтування (на прикладі флеш-носія, спільного каталогу віртуальної машини тощо). Уміти інтерпретувати вміст каталогу /dev. Одержувати та аналізувати відомості про використання дискового простору файловою системою (команда df). Уміти виводити та аналізувати таблицю розділів за допомогою команди fdisk. Уміти створювати файлові архіви у форматах gzip та bzip2, переглядати вміст цих архівів та здійснювати їх розархівування. Уміти пояснювати відмінність між блоковими та символьними пристроями. Уміти описувати етапи, з яких складається процес архівування у Linux. Знати найвідоміші формати файлових архівів у Linux та окреслювати коло ситуацій, коли використання кожного із них доцільне.

Лабораторна робота	Потребує теоретичних відомостей з тем	Засоби віртуалізації, інше ПЗ	Результати навчання
Лабораторна робота №11. Програмування роботи з файлами у Windows	Початкові відомості про ОС. Процеси та потоки. Пам'ять. Файлові системи	без віртуалізації, Code::Blocks Amazon EC2, Code::Blocks	Уміти здійснювати пошук довідкової інформації щодо використання Win32 API в MSDN Library. Уміти використовувати API-функції CreateFile(), CloseHandle(), SetFilePointer(), ReadFile(), WriteFile(), GetFileInformationByHandle(), DeleteFile(), CopyFile(), MoveFile(), CreateDirectory(), RemoveDirectory(), GetCurrentDirectory().
Лабораторна робота №12. Програмування роботи з файлами у Linux	Початкові відомості про ОС. Процеси та потоки. Пам'ять. Файлові системи.	VirtualBox, g++ VirtualBox, g++, Code::Blocks Amazon EC2, g++	Уміти здійснювати пошук довідкової інформації щодо використання бібліотечних функцій стандарту POSIX у документації man. Уміти використовувати функції open(), close(), lseek(), read(), write(), fstat(), rename(), mkdir(), ulink(), rmdir().
Лабораторна робота №13. Користувачі та групи у Linux	Початкові відомості про ОС. Процеси та потоки. Пам'ять. Файлові системи. Безпека	VirtualBox Amazon EC2 без віртуалізації	Уміти пояснювати призначення та інтерпретувати вміст конфігураційних файлів /etc/passwd, /etc/group, /etc/shadow. Уміти створювати та вилучати облікові записи користувачів за допомогою команд adduser, userdel. Уміти створювати та вилучати групи, додавати користувача до групи за допомогою команд groupadd, groupdel, usermod. Уміти змінювати паролі користувачів за допомогою команди passwd. Уміти змінювати власника та права доступу до файлів за допомогою команд chown, chmod. Уміти пояснювати атрибути файлів та каталогів, які виводяться командою ls -l, у тому числі рядок повноважень. Уміти здійснювати операції з обліковим записом за допомогою вікна Параметри користувачів. Уміти здійснювати операції з групами за допомогою вікна Параметри груп.
Лабораторна робота №14. Користувачі та групи у Windows	Початкові відомості про ОС. Процеси та потоки. Пам'ять. Файлові системи. Безпека	Amazon EC2 без віртуалізації	Уміти здійснювати операції з користувачами та групами за допомогою оснащення Управління комп'ютером (Computer Management). Уміти встановлювати дозволи для файлів та каталогів засобами файлового менеджера Провідник (Explorer). Уміти пояснювати суть основних та спеціальних дозволів для файлів та каталогів у Windows.

Додаток Л.

ОПИС ЧОТИРИРІВНЕВОЇ СТРУКТУРИ, ЩО ЛЕЖИТЬ В ОСНОВІ МЕТОДИКИ ЗАСТОСУВАННЯ ТЕХНОЛОГІЙ ВІРТУАЛІЗАЦІЇ UNIX-ПОДІБНИХ ОС У ПІДГОТОВЦІ БАКАЛАВРІВ ІНФОРМАТИКИ

Рівень I. Мета.

Рівень II. Завдання.

Рівень III. Компетентності.

Рівень IV. Результати навчання.

Мета застосування технологій віртуалізації Unix-подібних ОС у підготовці бакалаврів інформатики полягає у створенні умов для формування в студентів інформатичних компетентностей щодо ОС.

*Символом * позначено результати навчання, для досягнення яких може бути найбільш активно застосовано технологій віртуалізації.*

Завдання 1. Навчання основ теорії ОС

Компетентності та відповідні результати навчання:

- Знати та розуміти основні етапи, напрями та тенденції розвитку ОС
 - Орієнтуватися в історії розвитку ОС і на її основі ілюструвати взаємозв'язок між розвитком апаратного та програмного забезпечення
 - Уміти аналізувати спадковість тенденцій у проектуванні ОС
 - Знати історію виникнення ОС Linux
- Орієнтуватися у класифікації ОС, з урахуванням їх архітектури та сфери застосування
 - Знати, за яких умов ОС вважається мережною
 - Знати відмінності розподілених ОС від традиційних ОС
 - Орієнтуватися в архітектурах ОС (монолітні, багаторівневі, модульні, мікроядра), знати переваги та недоліки кожної з цих архітектур
 - Уміти характеризувати ОС залежно від сфери їх застосування (від ОС мейнфреймів до ОС смарт-карт)
 - Уміти розрізняти поняття ядра Linux, дистрибутиву Linux та випуску Linux
- Знати та розуміти теоретичні основи будови та функціонування ОС
 - Уміти пояснювати роль ОС у взаємодії користувача з комп'ютером та у керуванні системними ресурсами
 - Уміти пояснювати роль абстракцій у роботі ОС
 - Уміти порівнювати режим ядра та режим користувача ОС

- Уміти аналізувати роль ієрархії процесів у Unix-подібних ОС та у Windows
- Уміти характеризувати моделі ОС, залежно від кількості у ній процесів та потоків
- Уміти обґрунтовувати потребу у багатозадачності у сучасних ОС
- Уміти пояснювати відмінність псевдопаралелізму від реального паралелізму у багатопроесорних системах
- Уміти характеризувати типові стани процесів, а також умови та особливості переходу процесу між цими станами
- Уміти порівнювати системні виклики стандарту POSIX та виклики інтерфейсу прикладного програмування Win32 API
- Уміти описувати призначення основних типових структур даних для зберігання відомостей про процеси та потоки
- Уміти порівнювати особливості реалізації процесів та потоків у Linux та Windows
- Уміти описувати ієрархію пам'яті комп'ютера, тенденції щодо вартості та обсягів різних видів пам'яті, а також швидкості доступу до них
- Уміти характеризувати основні підходи до організації пам'яті, переваги та недоліки кожного з них
- Уміти пояснювати суть явища фрагментації, відмінності зовнішньої й внутрішньої фрагментації
- Уміти обґрунтовувати потребу в механізмах кешування, підкачування та пояснювати їх суть
- Розуміти суть основних алгоритмів заміщення сторінок, їх переваги та недоліки
- Знати типові файлові атрибути
- Уміти порівнювати правила іменування файлів у файлових системах для Unix-подібних ОС та файлових системах для Windows
- Уміти описувати призначення дискових розділів та їх роль у завантаженні ОС
- Уміти характеризувати основи будови та функціонування конкретних файлових систем (FAT, NTFS, UFS, ext)
- Уміти обґрунтовувати потребу та пояснювати суть технології журналювання
- Уміти пояснювати роль віртуальних файлових систем у реалізації сумісності файлових систем
- Уміти визначати потенційні загрози для ОС
- Уміти пояснювати зв'язки та відмінності між процедурами ідентифікації, аутентифікації, авторизації
- Уміти характеризувати типи аутентифікації (локальна, мережна), її форми (із використанням паролю, фізичного об'єкту, біометричних параметрів), описувати конкретні варіанти аутентифікації (на основі односторонніх функцій, одноразових паролів тощо)

- Уміти описувати основні компоненти системи безпеки у Linux та у Windows
- Уміти пояснювати суть принципу мінімальних повноважень
- Уміти порівнювати авторизацію на основі списків керування доступом та авторизацію за допомогою переліків можливостей
- Уміти характеризувати основні особливості авторизації у Linux та Windows
- Уміти обґрунтувати необхідність аудиту в ОС
- Уміти описувати основи організації аудиту у Linux та Windows
- Уміти характеризувати основні вимоги до планування
- Уміти пояснювати залежність вибору об'єкта планування в ОС від моделі реалізації потоків
- Уміти пояснювати відмінність між процесами, обмеженими швидкістю обчислень, та процесами, обмеженими швидкістю роботи пристроїв введення-виведення
- Уміти порівнювати планування без витіснення та з витісненням
- Уміти характеризувати підходи до планування залежно від сфери застосування (системи пакетної обробки, інтерактивні системи, системи реального часу) та описувати суть роботи основних алгоритмів для кожної із цих сфер
- Уміти описувати основи планування у Linux та Windows (об'єкт планування, алгоритми планування, система пріоритетів)
- Уміти обґрунтовувати потребу у міжпроцесовій та міжпотоківій взаємодії
- Уміти наводити приклади потенційних проблем, які можуть виникати під час виконання паралельних операцій кількома різними задачами
- Уміти характеризувати основні синхронізаційні примітиви (семафори, м'ютекси, умовні змінні, монітори, бар'єри тощо)
- Уміти характеризувати основні механізми передачі даних між процесами (сигнали, спільно використовувана пам'ять, повідомлення, відображувана пам'ять)
- Уміти орієнтуватися у різних тлумаченнях терміну «віртуалізація»*
- Уміти пояснювати місце віртуалізації в ОС
- Уміти характеризувати основні напрями (віртуальні машини, віртуальні контейнери, віртуалізація програмних застосунків, віртуалізація робочого столу) та методи (повна емуляція, емуляція системних бібліотек, паравіртуалізація, динамічна трансляція, апаратна віртуалізація) віртуалізації*
- Уміти аналізувати доцільність вибору засобу віртуалізації залежно від сфери застосування, наявних ресурсів та вимог
- Уміти пояснювати зв'язок між технологіями віртуалізації та хмарними технологіями*

Завдання 2. Освоєння основ адміністрування ОС

Компетентності та відповідні результати навчання:

- Вміти використовувати інтерфейс користувача Unix-подібних ОС (графічний інтерфейс, інтерфейс командного рядка)
 - Орієнтуватися у віконних системах для Linux та окреслювати коло ситуацій, коли потрібний графічний інтерфейс
 - Уміти пояснювати суть та призначення механізму `sudo`*
 - Орієнтуватися в основних елементах робочого столу та меню Linux (на прикладі Xubuntu Linux)*
 - Уміти здійснювати навігацію та найпростіші операції з файлами у графічному файловому менеджері (на прикладі Thunar)*
 - Уміти аналізувати вміст запрошення командного рядка*
 - Уміти користуватися буфером хронології команд*
 - Уміти виводити вміст каталогу, у тому числі відформатований та відфільтрований, за допомогою команди `ls`*
 - Уміти виводити вміст файлу за допомогою команди `cat`*
 - Володіти прийомами посторінкового виведення (команди `more`, `less`)*
 - Уміти здійснювати пошук у виводі команд за ключовим словом*
 - Уміти здійснювати просту конвеєризацію двох команд та перескєрування виводу команди у файл*
 - Уміти одержувати та використовувати довідку у форматі `man` (за допомогою команди та на веб-ресурсах)*
 - Уміти одержувати та використовувати довідку за допомогою команди `help`*
 - Уміти виконувати команди з правами адміністратора*
 - Уміти завершувати роботу засобами графічного інтерфейсу*
 - Уміти завершувати роботу за допомогою команди `shutdown`*
 - Уміти здійснювати основні операції у простому текстовому редакторі з графічним інтерфейсом (на прикладі Mousepad)*
 - Уміти здійснювати основні операції у простому текстовому редакторі з інтерфейсом командного рядка (на прикладі nano)*
- Вміти здійснювати налаштування мережного з'єднання у Unix-подібних ОС
 - Уміти пояснювати призначення конфігураційних файлів та окреслювати коло ситуацій, коли доцільно редагувати їх вручну
 - Уміти перевіряти роботу мережного з'єднання шляхом пінгування зовнішньої адреси*
 - Уміти інтерпретувати вміст файлу `/etc/network/interfaces`*
 - Відповідно до синтаксису файлу `/etc/network/interfaces`, вміти описувати простий інтерфейс Ethernet (методи DHCP та static) та loopback-інтерфейс*

- Уміти характеризувати призначення віртуального мережного loopback-інтерфейсу
- Уміти одержувати та інтерпретувати відомості про мережні інтерфейси за допомогою команди `ifconfig`*
- Уміти встановлювати статичну конфігурацію для інтерфейсу Ethernet за допомогою команди `ifconfig`*
- Уміти встановлювати динамічну конфігурацію для інтерфейсу Ethernet за допомогою команди `dhclient`*
- Уміти одержувати та інтерпретувати відомості про наявні мережні з'єднання та створювати нові засобами графічного інтерфейсу (наприкладі GUI-засобу Мережні з'єднання)*
- Уміти підключатися до мережі та відключитися від неї засобами графічного інтерфейсу*
- Уміти здійснювати налаштування з'єднання через проксі-сервер шляхом редагування конфігураційних файлів*
- Вміти встановлювати програмне забезпечення у Unix-подібних ОС, виконувати його наступне оновлення та деінсталяцію
 - Уміти здійснювати встановлення Linux за допомогою мережного інсталятора*
 - Уміти пояснювати роль пакунків та репозиторіїв у встановленні ПЗ у Linux
 - Уміти описувати особливості організації офіційних репозиторіїв Ubuntu
 - Орієнтуватися у найпоширеніших форматах пакунків
 - Уміти порівнювати інструменти для роботи з пакунками у deb-сумісних дистрибутивах, зокрема з точки зору автоматичного врахування залежностей*
 - Уміти встановлювати та вилучати пакунки, а також одержувати відомості про вже встановлені пакунки за допомогою команди `dpkg`*
 - Уміти пояснювати призначення конфігураційних файлів та окреслювати коло ситуацій, коли доцільно редагувати їх вручну
 - Уміти аналізувати та редагувати вміст файлу `/etc/apt/sources.list`*
 - Уміти встановлювати та вилучати пакунки, здійснювати пошук пакунків у підключених репозиторіях, автоматичне вилучення непотрібних пакунків та deb-файлів за допомогою утиліти `apt`*
 - Уміти встановлювати та вилучати пакунки, а також здійснювати пошук пакунків у репозиторіях за допомогою менеджера пакунків Synaptic*
 - Уміти аналізувати та змінювати налаштування утиліти `apt` за допомогою GUI-засобу Програмне забезпечення та оновлення*
- Вміти відстежувати роботу процесів та потоків в ОС і керувати ними
 - Уміти характеризувати стани процесів у Linux
 - Уміти пояснювати роль рівня пріоритету у Linux

- Уміти одержувати та інтерпретувати характеристики процесів Linux за допомогою Диспетчера задач*
- Уміти запускати, призупиняти, поновлювати та завершувати процеси Linux засобами Диспетчера задач, утиліти `top`*
- Уміти одержувати та інтерпретувати відомості про процеси, у тому числі відфільтровані, за допомогою команди `ps`*
- Уміти здійснювати звичайне і примусове завершення процесів за допомогою команд `kill` та `killall`*
- Уміти запускати процеси Linux у фоновому режимі*
- Уміти змінювати пріоритет процесу засобами команд `nice` та `renice`*
- Уміти здійснювати завершення та зміну пріоритету процесу засобами утиліти `top`*
- Уміти характеризувати типи запуску та стани служб у Windows*
- Уміти одержувати та інтерпретувати характеристики процесів (задач та процесів) за допомогою Диспетчера задач*
- Уміти завершувати процеси за допомогою Диспетчера задач*
- Уміти одержувати та інтерпретувати характеристики служб за допомогою оснащення Служби*
- Уміти зупиняти та запускати служби, змінювати тип служби за допомогою оснащення Служби*
- Уміти запускати програми Windows з командного рядка*
- Уміти виводити відомості, у тому числі відформатовані, про процеси за допомогою команди `tasklist`*
- Уміти завершувати процеси за допомогою команди `taskkill`*
- Вміти здійснювати моніторинг використання різних видів пам'яті в ОС
 - Уміти одержувати та аналізувати відомості про використання пам'яті у Linux за допомогою команди `free`*
 - Уміти одержувати та аналізувати статистику використання віртуальної пам'яті у Linux за допомогою команди `vmstat`*
 - Уміти визначати, який саме блоковий файл відповідає розділу підкачування у Linux*
 - Уміти одержувати та аналізувати відомості про використання пам'яті та часу ЦП у Windows за допомогою Диспетчера задач*
 - Уміти показувати розташування файлу підкачування у Windows*
- Володіти основами роботи з файловою системою у Unix-подібних ОС
 - Уміти виводити вміст каталогу, у тому числі відформатований та відфільтрований, за допомогою команди `ls`*
 - Уміти виводити вміст файлу за допомогою команди `cat`*
 - Уміти здійснювати основні операції у простому текстовому редакторі з графічним інтерфейсом (наприкладі `Mousepad`)*
 - Уміти здійснювати основні операції у простому текстовому редакторі з інтерфейсом командного рядка (наприкладі `nano`)*
 - Уміти здійснювати навігацію та найпростіші операції з файлами у графічному файловому менеджері (наприкладі `Thunar`)*

- Уміти здійснювати навігацію каталогами за допомогою команди `cd`*
- Уміти створювати порожній файл засобами команди `touch`*
- Уміти створювати нові каталоги за допомогою команди `mkdir`*
- Уміти копіювати, переміщувати, перейменовувати, видаляти файли та каталоги за допомогою команд `cp`, `mv`, `rm`*
- Уміти пояснювати суть механізму монтування
- Уміти здійснювати автоматичне/ручне монтування та розмонтування (на прикладі флеш-носія, спільного каталогу віртуальної машини тощо)*
- Уміти показувати розташування файлу підкачування у Windows*
- Уміти інтерпретувати вміст каталогу `/dev`*
- Уміти одержувати та аналізувати відомості про використання дискового простору файловою системою (команда `df`)*
- Уміти виводити та аналізувати таблицю розділів за допомогою команди `fdisk`*
- Уміти створювати файлові архіви у форматах `gzip` та `bzip2`, переглядати вміст цих архівів та здійснювати їх розархівування*
- Уміти пояснювати відмінність між блоковими та символьними пристроями
- Уміти описувати етапи, з яких складається процес архівування у Linux
- Знати найвідоміші формати файлових архівів у Linux та окреслювати коло ситуацій, коли використання кожного із них доцільне
- Володіти основами роботи з налаштуваннями системи безпеки ОС
 - Уміти пояснювати призначення та інтерпретувати вміст конфігураційних файлів `/etc/passwd`, `/etc/group`, `/etc/shadow`*
 - Уміти створювати та вилучати облікові записи користувачів за допомогою команд `adduser`, `userdel`*
 - Уміти створювати та вилучати групи, додавати користувача до групи за допомогою команд `groupadd`, `groupdel`, `usermod`*
 - Уміти змінювати паролі користувачів за допомогою команди `passwd`.
 - Уміти змінювати власника та права доступу до файлів за допомогою команд `chown`, `chmod`*
 - Уміти пояснювати атрибути файлів та каталогів, які виводяться командою `ls -l`, у тому числі рядок повноважень*
 - Уміти здійснювати операції з обліковим записами за допомогою вікна Параметри користувачів*
 - Уміти здійснювати операції з групами за допомогою вікна Параметри груп*

- Уміти здійснювати операції з користувачами та групами за допомогою оснащення Управління комп'ютером (Computer Management)*
- Уміти встановлювати дозволи для файлів та каталогів засобами файлового менеджера Провідник (Explorer)*
- Уміти пояснювати суть основних та спеціальних дозволів для файлів та каталогів у Windows*

Завдання 3. Навчання основ системного програмування

Компетентності та відповідні результати навчання:

- Володіти основами системного програмування мовою C++ у Linux (у Linux і Windows)
 - Уміти здійснювати пошук довідкової інформації щодо використання Win32 API в MSDN Library
 - Уміти характеризувати призначення основних типів даних Windows
 - Пояснювати суть та призначення угорської нотації, характеризувати її місце серед інших підходів до формування ідентифікаторів
 - Уміти використовувати API-функції CreateProcess(), CloseHandle(), CreateThread(), WaitForSingleObject(), WaitForMultipleObjects(), GetExitCodeProcess()*
 - Уміти здійснювати пошук довідкової інформації щодо використання бібліотечних функцій стандарту POSIX у документації man
 - Уміти пояснювати механізм роботи технології fork+exec
 - Уміти компілювати програмний код засобами компілятора g++ з командного рядка (у тому числі, з підключенням бібліотеки pthread)*
 - Уміти запускати на виконання відкомпільований файл програми засобами командного рядка*
 - Уміти використовувати функції fork(), execve(), waitpid(), exit(), pthread_create(), pthread_join()*
 - Уміти використовувати API-функції CreateFile(), CloseHandle(), SetFilePointer(), ReadFile(), WriteFile(), GetFileInformationByHandle(), DeleteFile(), CopyFile(), MoveFile(), CreateDirectory(), RemoveDirectory(), GetCurrentDirectory()
 - Уміти використовувати функції open(), close(), lseek(), read(), write(), fstat(), rename(), mkdir(), unlink(), rmdir()

Завдання 4. Створення передумов для наступної самоосвіти з ОС

Компетентності та відповідні результати навчання:

- Вміти одержувати довідкові відомості щодо використання ОС та програмування у них

- Уміти одержувати та використовувати довідку у форматі man (за допомогою команди та на веб-ресурсах)*
- Уміти одержувати та використовувати довідку за допомогою команди help*
- Уміти здійснювати пошук довідкової інформації щодо використання Win32 API в MSDN Library
- Уміти здійснювати пошук довідкової інформації щодо використання бібліотечних функцій стандарту POSIX у документації man

Додаток М.

**ЗРАЗОК АНКЕТИ ДЛЯ ЕКСПЕРТНОГО ОПИТУВАННЯ
З МЕТОЮ ВИЗНАЧЕННЯ ЗМІСТОВОГО МОДУЛЯ
ДЛЯ ОПИСУ ФОРМ, МЕТОДІВ ТА ЗАСОБІВ НАВЧАННЯ ПІД ЧАС
ЗАСТОСУВАННЯ ТЕХНОЛОГІЙ ВІРТУАЛІЗАЦІЇ UNIX-ПОДІБНИХ ОС
У ПІДГОТОВЦІ БАКАЛАВРІВ ІНФОРМАТИКИ**

Пріоритетність тем курсу з операційних систем

Добрий день, шановні колеги! Просимо Вас дати відповіді на кілька запитань. Своєю участю у цьому опитуванні Ви допоможете дослідженню застосування технологій віртуалізації unix-подібних операційних систем у підготовці бакалаврів інформатики. Заздалегідь вдячні.

***Обов'язкове поле**

Вкажіть назву навчального закладу, у якому Ви працюєте: *

Ваша відповідь _____

Ви маєте досвід викладання операційних систем студентам...

☐ інженерних спеціальностей

☐ педагогічних спеціальностей

☐ Інше: _____

Оцініть важливість запропонованих тем курсу з операційних систем для бакалаврів інформатики (майбутні вчителі інформатики): *

	3 (обов'язково розглядати)	2 (варто розглядати)	1 (можна розглядати)
Початкові відомості про операційні системи	☐	☐	☐
Процеси та потоки	☐	☐	☐
Пам'ять	☐	☐	☐
Файлові системи	☐	☐	☐
Безпека	☐	☐	☐
Планування	☐	☐	☐
Міжпроцесова та міжпоточкова взаємодія	☐	☐	☐
Віртуалізація	☐	☐	☐
Керування пристроями	☐	☐	☐

Якщо Ви маєте коментарі до вибору, зробленого у попередньому запитанні, вкажіть їх тут, будь ласка:

Ваша відповідь

НАДІСЛАТИ

Ніколи не вказуйте паролі в Google Формах.

Додаток Н.

ОПРАЦЮВАННЯ РЕЗУЛЬТАТІВ ЕКСПЕРТНОГО ОПИТУВАННЯ
ЗАСОБАМИ МОВИ RВхідні дані

Файл os_course_units.txt

Rater1	Rater2	Rater3	Rater4	Rater5	Rater6	Rater7	Rater8	Rater9	Rater10	Rater11	Rater12	Rater13
3	3	3	3	3	3	1	3	3	2	3	3	3
2	3	2	2	3	2	2	2	2	2	3	2	3
2	3	2	2	3	2	2	3	3	3	3	3	3
3	3	3	3	3	2	2	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	1	3	3	3
2	2	3	3	2	2	1	2	2	1	3	2	3
1	1	2	2	2	1	2	2	3	2	3	2	3
2	1	3	3	2	1	3	2	2	2	3	2	3
3	3	3	3	2	2	2	3	3	3	2	2	2

Програмний код (значущість – за критерієм хі-квадрат)

dt <- read.table("os_course_units.txt", header = TRUE)

library(DescTools)

KendallW(dt, correct=TRUE, test=TRUE)

Результати

Kendall's coefficient of concordance Wt

data: dt

Kendall chi-squared = 30.325, df = 8, subjects = 9, raters = 13,

p-value = **0.0001851**

alternative hypothesis: Wt is greater 0

sample estimates:

Wt

0.2915888Програмний код (значущість – за перестановочним тестом)

library("synchrony")

wr = kendall.w(dt, nrands=100000)

wr

Результати

Kendall's W (uncorrected for ties): 0.2

Kendall's W (corrected for ties): **0.2916**

Spearman's ranked correlation: 0.2326

Kendall's W p-value (one-tailed test [greater]): **7e-05**

Додаток П.

ТЕСТОВЕ ОПИТУВАННЯ «UNIX-ПОДІБНІ ОПЕРАЦІЙНІ СИСТЕМИ»

Unix-подібні операційні системи

Контрольне опитування з теми "Unix-подібні операційні системи" для студентів напряму підготовки 6.040302. Інформатика*

Введіть Ваше прізвище та ім'я: *

Ваша відповідь

Введіть номер Вашої групи: *

Ваша відповідь

Виберіть правильний варіант відповіді

До Unix-подібних ОС належить ОС...

- ☐ MS DOS
- ☐ MS Windows
- ☐ Linux

GNOME, KDE, Xfce є назвами...

- ☐ форматів інсталяційних пакунків для Unix-подібних ОС
- ☐ Unix-подібних ОС
- ☐ графічних інтерфейсів для Unix-подібних ОС

Системні виклики у Unix-подібних ОС регламентуються стандартом...

- ☐ MULTIX
- ☐ UNIX
- ☐ POSIX

Програма, що працює у цьому режимі, одержує повний доступ до апаратного забезпечення і може використати будь-яку інструкцію, котру здатних виконати даний комп'ютер. Цей режим називають...

- ☐ режимом користувача
- ☐ режимом ядра
- ☐ командним режимом
- ☐ віртуальним режимом

PID - це...

- ☐ ідентифікатор користувача
- ☐ ідентифікатор процесу
- ☐ ідентифікатор пакунку

Фонові процеси у Unix-подібних операційних системах мають назву...

- ☐ зомбі
- ☐ демони
- ☐ орки

Символ *r* (англійська *ar*) у рядку повноважень файлу означає, що...

- ☐ деякому суб'єкту дозволено читання цього файлу
- ☐ деякому суб'єкту дозволено запис цього файлу
- ☐ деякому суб'єкту дозволено виконання цього файлу

Команда `touch` призначена для...

- ☐ вилучення файлу
- ☐ перейменування файлу
- ☐ створення нового файлу

X Windows System (X11) - це назва...

- ☐ віконної системи для Unix-подібних ОС
- ☐ одного з дистрибутивів Linux
- ☐ емулятора терміналу у Unix-подібних ОС

У Unix-подібних ОС що більше числове значення відповідає пріоритету процесу, то...

- ☐ нижчим є сам пріоритет
- ☐ вищим є сам пріоритет

Цей процес у Unix-подібних системах створюється першим під час ініціалізації ОС. Ідеться про процес...

- ☐ bash
- ☐ cron
- ☐ init

Процес, що перебуває у цьому стані, вже фактично завершений, але його структуру даних ще не вилучено з пам'яті. Йдеться про процес у стані...

- ☐ очікування (sleeping)
- ☐ зомбі (zombie)
- ☐ мертвий (dead)
- ☐ зупинений (stopped)

Набір адрес, до яких процес має доступ, називають...

- ☐ адресним простором
- ☐ деревом процесів
- ☐ потоком виконання

Створення нового процесу у Unix-подібних ОС здійснюється поєднанням системних викликів...

- ☐ fork + waitpid
- ☐ exec + waitpid
- ☐ fork + exec
- ☐ exec + exit

Суть технології підкачування (свопінгу) полягає у тому, що...

- ☐ з метою пришвидшення роботи системи здійснюється кешування сторінок, які використовуються найчастіше

- ☐ з метою скорочення часу реагування системи на запити користувача системні процес запускаються з порівняно низьким пріоритетом
- ☐ з метою економії оперативної пам'яті частина сторінок тимчасово вивантажується на жорсткий диск

В Unix-подібних операційних системах відомості про користувачів традиційно зберігаються у файлі...

- ☐ /etc/shadow
- ☐ /etc/passwd
- ☐ /etc/users
- ☐ /etc/group

Поле UID у виводі команди ps (рис. 1) - це...

- ☐ ідентифікатор даного процесу
- ☐ ідентифікатор користувача, який створив даний процес
- ☐ пріоритет процесу

```
student@lab01:~$ ps -l
```

F	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	TTY	TIME	CMD
0	S	1000	3099	3097	0	80	0	-	1462	wait	pts/1	00:00:01	bash
0	R	1000	3134	3099	0	80	0	-	619	-	pts/1	00:00:00	ps

Рис. 1

У прикладі на рис. 2 команда mv виконує...

- ☐ вилучення файлу
- ☐ переміщення файлу
- ☐ перейменування файлу

```
student@lab01:~$ mv ~/cat2/file1 ~/cat2/file4
```

Рис. 2

Шлях, що складається з єдиного символу /, відповідає...

- ☐ каталогу, в якому зберігаються тимчасові файли
- ☐ кореневому каталогу
- ☐ домашньому каталогу

Команда `renice` на рис. 3...

- ☐ знижує на 1 пріоритет процесу з PID 3475
- ☐ підвищує на 1 пріоритет процесу з PID 3475
- ☐ встановлює пріоритет 3475 для процесу з PID 1
- ☐ встановлює пріоритет -1 для процесу з PID 3475

```
student@lab01:~$ sudo renice -n -1 -p 3475
```

Рис. 3

Програма створює дочірній процес та запускає у ньому програму `helloworld`. Ці дії реалізує фрагмент коду (рис. 4)...

- ☐ фрагмент а)
- ☐ фрагмент б)
- ☐ фрагмент в)

```
а) pid_t pid;
   int res;
   pid = fork();
   if (pid == 0) res = execve("/home/student/helloworld",0,0);

б) pid_t pid;
   int res;
   fork(pid);
   if (pid == 0) res = execve("/home/student/helloworld",0,0);

в) pid_t pid;
   int res;
   fork(&pid);
   if (pid == 0) res = execve("/home/student/helloworld",0,0);
```

Рис. 4

У Unix-подібних ОС використовується розмежування доступу, згідно з яким...

- ☐ для кожного ресурсу (файлу, каталогу тощо) задано список суб'єктів (користувачів, груп), котрим дозволено використовувати цей ресурс
- ☐ для кожного суб'єкта (користувача, групи) задано список ресурсів (файлів, каталогів тощо), котрі йому дозволено використовувати

Користувачу потрібні права адміністратора...

- ☐ для штучного пониження пріоритету власного процесу користувачу потрібні права адміністратора

- ☐ для штучного підвищення пріоритету власного процесу користувачу потрібні права адміністратора
- ☐ обидва вказані вище варіанти
- ☐ жодний із вказаних вище варіантів

У наведеному на рис. 5 фрагменті коду...

- ☐ створюється новий процес; дескриптор процесу поміщається в масив threads; процес матиме ім'я fun1; аргумент для функції main новоствореного процесу братиметься за адресою &arg
- ☐ створюється новий потік; вказівник на адресу цієї функції зберігається у threads[0]; дескриптор потоку поміщається за адресою у змінну fun1; аргументи для функції братимуться за адресою &arg
- ☐ створюється новий потік; дескриптор потоку поміщається в масив threads; у потоці виконуватиметься функція fun1; аргументи для функції братимуться за адресою &arg

```
er = pthread_create(&threads[0], NULL, fun1, (void *) & arg);
```

Рис. 5

Виберіть усі правильні варіанти відповіді (один або декілька)

До дистрибутивів Linux належить (належать)...

- ☐ Ubuntu
- ☐ Unity
- ☐ Fedora

Unix-подібні ОС використовують файлову систему (файлові системи)...

- ☐ ext4
- ☐ ext3
- ☐ ext2
- ☐ NTFS

Linux-системи використовують формат (формати) пакунків...

- ☐ deb
- ☐ conf
- ☐ rpm

У файлових системах на базі UFS...

- ☐ символи верхнього і нижнього регістрів в іменах файлів не розрізняються
- ☐ розширення файлів не виконує суттєвої ролі, нерідко файли не мають жодного розширення
- ☐ у повних іменах файлів використовуються прямі слеші (/)

Рядок повноважень деякого файлу має вигляд `-rw-rw-r--`. Це дає підстави стверджувати, що...

- ☐ власникові файлу дозволено читання цього файлу
- ☐ власникові файлу дозволено запис цього файлу
- ☐ власникові файлу дозволено виконання цього файлу

Завершіть речення

Стандартний формат документації для Unix-подібних ОС та однойменна команда для доступу до документації у такому форматі має назву...

Ваша відповідь

Коли процес А породжує процес В, то процес А називають батьківським, а процес В - ...

Ваша відповідь

Множина облікових записів, об'єднаних з метою встановлення для них спільних налаштувань, зокрема параметрів доступу, має назву...

Ваша відповідь

Заповніть пропуски

Для встановлення пакунку `mc` можна скористатися командою `sudo apt-get _____ mc`

Ваша відповідь

Додаток Р.

ОПИТУВАЛЬНИК ДЛЯ ВИКЛАДАЧІВ ДЛЯ ДОСЛІДЖЕННЯ ДОСВІДУ НАВЧАННЯ ОС У ЗАКЛАДАХ ВИЩОЇ ОСВІТИ УКРАЇНИ



Вивчення досвіду навчання UNIX-подібних операційних систем у закладах вищої освіти України

Добрий день, шановні колеги!

Пропонуємо до Вашої уваги опитування, головне завдання якого - вивчити те, як під час навчання UNIX-подібних операційних систем можуть бути використані технології віртуалізації.

Та водночас, запускаючи це опитування, ми також ставили перед собою й інше завдання - дослідити вітчизняний досвід викладання "Операційних систем та системного програмування" (чи інших подібних дисциплін) загалом, тож в анкеті є й запитання, безпосередньо не пов'язані з технологіями віртуалізації.

Анкетування адресоване викладачам "Операційних систем", "Системного програмування", "Операційних систем та системного програмування" та інших подібних дисциплін.

Будемо дуже вдячні Вам за відповіді на запитання цієї анкети. Сподіваємося разом із Вами зробити викладання "Операційних систем та системного програмування" в українських вишах кращим, а нашу освіту - якіснішою.

Це анкетування анонімне. Відповіді збираються і надсилаються нам автоматично та без прив'язки до облікового запису Google, адреси електронної пошти, IP-адреси чи інших персональних даних респондента (лише Ваші відповіді, а також дата і час їх надсилання). У разі, коли самі відповіді на окремі запитання анкети міститимуть Ваші персональні дані, зауважимо, що будь-яка їх публікація можлива лише після попереднього одержання Вашої особистої згоди.

***Обов'язкове поле**

1. Як називається дисципліна (дисципліни), у межах якої (яких) Ваші студенти вивчають операційні системи? *

- ☐ "Операційні системи та системне програмування"
- ☐ "Операційні системи"
- ☐ "Системне програмування"
- ☐ Інше:

2. Студенти якого напрямку (напрямів) вивчають дисципліну, про яку (які) йшлося у попередньому запитанні? *

- ☐ 6.040301. Прикладна математика
- ☐ 6.040302. Інформатика*
- ☐ 6.040303. Системний аналіз
- ☐ 6.050101. Комп'ютерні науки
- ☐ 6.050102. Комп'ютерна інженерія
- ☐ 6.050103. Програмна інженерія
- ☐ 6.050201. Системна інженерія
- ☐ 6.050202. Автоматизація та комп'ютерно-інтегровані технології
- ☐ 6.170101. Безпека інформаційних і комунікаційних систем
- ☐ 6.170102. Системи технічного захисту інформації
- ☐ 6.170103. Управління інформаційною безпекою
- ☐ Інше:

3. З якою операційною системою (операційними системами, надалі - ОС) працюють Ваші студенти на лабораторних заняттях з навчальної дисципліни "Операційні системи та системне програмування" чи іншої подібної дисципліни? *

- ☐ Linux
- ☐ Windows
- ☐ FreeBSD
- ☐ Minix
- ☐ OpenSolaris
- ☐ OpenBSD
- ☐ Інше:

4. Якщо під час лабораторних занять Ваші студенти працюють з ОС Linux, то який саме дистрибутив (дистрибутиви) використовується? *

- ☐ Linux Mint
- ☐ Ubuntu Linux (у тому числі, Kubuntu, Xubuntu, Lubuntu, Edubuntu тощо)
- ☐ Debian GNU/Linux
- ☐ Mageia
- ☐ Fedora
- ☐ open SUSE
- ☐ Cent OS
- ☐ PC Linux OS
- ☐ Arch Linux
- ☐ Slackware Linux
- ☐ Gentoo Linux
- ☐ не використовуємо Linux
- ☐ Інше:

5. Досвід використання яких програмних засобів віртуалізації для навчання ОС Ви маєте (навіть якщо якісь із цих засобів Ви застосовували у минулому і зараз не застосовуєте)? *

- ☐ VirtualBox
- ☐ Vmware Player
- ☐ Kernel Based Virtual Machine (KVM)
- ☐ OpenVZ
- ☐ FreeBSD jail
- ☐ Linux-Vserver
- ☐ Oracle VM Server for x86
- ☐ XenServer
- ☐ Proxmox VE
- ☐ Vmware Workstation
- ☐ Microsoft Hyper-V
- ☐ Vmware ESX
- ☐ Vmware Sphere
- ☐ Vmware Horizon
- ☐ Citrix View
- ☐ не маю досвіду використання програмних засобів віртуалізації
- ☐ Інше:

6. Які програмні засоби віртуалізації ви використовуєте для навчання ОС на даний момент? *

- ☐ VirtualBox
- ☐ Vmware Player
- ☐ Kernel Based Virtual Machine (KVM)
- ☐ OpenVZ
- ☐ FreeBSD jail
- ☐ Linux-Vserver
- ☐ Oracle VM Server for x86
- ☐ XenServer
- ☐ Proxmox VE
- ☐ Vmware Workstation
- ☐ Microsoft Hyper-V
- ☐ Vmware ESX
- ☐ Vmware Sphere
- ☐ Vmware Horizon
- ☐ Citrix View
- ☐ не використовую програмних засобів віртуалізації
- ☐ Інше:

7. Можливість застосування яких програмних засобів віртуалізації для навчання ОС Ви плануєте (обмірковуєте) на майбутнє? *

- ☐ VirtualBox
- ☐ Vmware Player
- ☐ Kernel Based Virtual Machine (KVM)
- ☐ OpenVZ
- ☐ FreeBSD jail
- ☐ Linux-Vserver
- ☐ Oracle VM Server for x86
- ☐ XenServer
- ☐ Proxmox VE
- ☐ Vmware Workstation
- ☐ Microsoft Hyper-V
- ☐ Vmware ESX
- ☐ Vmware Sphere
- ☐ Vmware Horizon
- ☐ Citrix View
- ☐ найближчим часом не планую використовувати програмних засобів віртуалізації
- ☐ Інше:

8. Як довго Ви використовуєте обраний Вами програмний засіб віртуалізації для навчання ОС?

Якщо Ви не використовуєте засобів віртуалізації, просто пропустіть це запитання. Якщо Ви маєте досвід використання з цією метою кількох засобів віртуалізації - будь ласка, вкажіть орієнтовну тривалість застосування кожного.

9. Які альтернативи засобам віртуалізації Ви застосовуєте під час навчання ОС? *

- ☐ є комп'ютерний клас, у якому ОС, що вивчається, є єдиною ОС
- ☐ ОС, що вивчається, встановлено паралельно з іншою ОС (мультизавантаження)
- ☐ не використовую жодних інших варіантів, окрім засобів віртуалізації
- ☐ Інше:

10. Чи підтримують процесори комп'ютерів, які Ви використовуєте для навчання ОС, технології апаратної віртуалізації (Intel VT, AMD-V)? *

- ☐ так, процесори з підтримкою такої технології встановлені на машинах у комп'ютерних класах, де я проводжу заняття
- ☐ так, процесор з підтримкою такої технології встановлений на сервері, на базі якого ми реалізуємо віддалений доступ до віртуалізованих ОС
- ☐ ні, процесорів з підтримкою такої технології у моєму розпорядженні немає
- ☐ важко відповісти
- ☐ Інше:

11. Оцініть важливість, яку має для Вас під час вибору програмних засобів віртуалізації наступний фактор: *

можливість формування компетентностей та умінь, визначених галузевим стандартом

- ☐ дуже важливо (5)
- ☐ важливо (4)
- ☐ не обов'язково, але доречно (3)
- ☐ не дуже важливо (2)
- ☐ взагалі не важливо (1)
- ☐ важко відповісти (0)

12. Оцініть важливість, яку має для Вас під час вибору програмних засобів віртуалізації наступний фактор: *

швидкість роботи

- ☐ дуже важливо (5)
- ☐ важливо (4)
- ☐ не обов'язково, але доречно (3)
- ☐ не дуже важливо (2)
- ☐ взагалі не важливо (1)
- ☐ важко відповісти (0)

13. Оцініть важливість, яку має для Вас під час вибору програмних засобів віртуалізації наступний фактор: *

надійність роботи

- ☐ дуже важливо (5)
- ☐ важливо (4)
- ☐ не обов'язково, але доречно (3)
- ☐ не дуже важливо (2)
- ☐ взагалі не важливо (1)
- ☐ важко відповісти (0)

14. Оцініть важливість, яку має для Вас під час вибору програмних засобів віртуалізації наступний фактор: *

можливість віртуалізації різних ОС (універсальність щодо гостей ОС)

- ☐ дуже важливо (5)
- ☐ важливо (4)
- ☐ не обов'язково, але доречно (3)
- ☐ не дуже важливо (2)
- ☐ взагалі не важливо (1)
- ☐ важко відповісти (0)

15. Оцініть важливість, яку має для Вас під час вибору програмних засобів віртуалізації наступний фактор: *

можливість встановлення програмного засобу віртуалізації поверх різних ОС (універсальність щодо основних ОС)

- ☐ дуже важливо (5)
- ☐ важливо (4)
- ☐ не обов'язково, але доречно (3)
- ☐ не дуже важливо (2)
- ☐ взагалі не важливо (1)
- ☐ важко відповісти (0)

16. Оцініть важливість, яку має для Вас під час вибору варіантів навчання ОС (У ТОМУ ЧИСЛІ Й СЕРЕД ВАРІАНТІВ БЕЗ ВІРТУАЛІЗАЦІЇ) наступний фактор: *

оперативність переключення між різними ОС

- ☐ дуже важливо (5)
- ☐ важливо (4)
- ☐ не обов'язково, але доречно (3)
- ☐ не дуже важливо (2)
- ☐ взагалі не важливо (1)
- ☐ важко відповісти (0)

17. Оцініть важливість, яку має для Вас під час вибору варіантів навчання ОС (У ТОМУ ЧИСЛІ Й СЕРЕД ВАРІАНТІВ БЕЗ ВІРТУАЛІЗАЦІЇ) наступний фактор: *

простота вилучення/заміни ОС, що вивчається

- ☐ дуже важливо (5)
- ☐ важливо (4)
- ☐ не обов'язково, але доречно (3)
- ☐ не дуже важливо (2)
- ☐ взагалі не важливо (1)
- ☐ важко відповісти (0)

18. Оцініть важливість, яку має для Вас під час вибору програмних засобів віртуалізації наступний фактор: *

можливість проведення у комп'ютерному класі занять з інших дисциплін

- ☐ дуже важливо (5)
- ☐ важливо (4)
- ☐ не обов'язково, але доречно (3)
- ☐ не дуже важливо (2)
- ☐ взагалі не важливо (1)
- ☐ важко відповісти (0)

19. Оцініть важливість, яку має для Вас під час вибору програмних засобів віртуалізації наступний фактор: *

простота організації студентської самостійної роботи з ОС, що вивчається

- ☐ дуже важливо (5)
- ☐ важливо (4)
- ☐ не обов'язково, але доречно (3)
- ☐ не дуже важливо (2)
- ☐ взагалі не важливо (1)
- ☐ важко відповісти (0)

20. Оцініть важливість, яку має для Вас під час вибору програмних засобів віртуалізації наступний фактор: *

унікнення надання студенту прав адміністратора

- ☐ дуже важливо (5)
- ☐ важливо (4)
- ☐ не обов'язково, але доречно (3)
- ☐ не дуже важливо (2)
- ☐ взагалі не важливо (1)
- ☐ важко відповісти (0)

21. Оцініть важливість, яку має для Вас під час вибору програмних засобів віртуалізації наступний фактор: *

підтримка процесорів як із вбудованою апаратною віртуалізацією, так і без неї

- ☐ дуже важливо (5)
- ☐ важливо (4)
- ☐ не обов'язково, але доречно (3)
- ☐ не дуже важливо (2)
- ☐ взагалі не важливо (1)
- ☐ важко відповісти (0)

22. Оцініть важливість, яку має для Вас під час вибору програмних засобів віртуалізації наступний фактор: *

можливість без значних організаційних зусиль впровадити програмний засіб у навчальний процес

- ☐ дуже важливо (5)
- ☐ важливо (4)
- ☐ не обов'язково, але доречно (3)
- ☐ не дуже важливо (2)
- ☐ взагалі не важливо (1)
- ☐ важко відповісти (0)

23. Оцініть важливість, яку має для Вас під час вибору програмних засобів віртуалізації наступний фактор: *

наявність у програмному засобі графічного інтерфейсу

- ☐ дуже важливо (5)
- ☐ важливо (4)
- ☐ не обов'язково, але доречно (3)
- ☐ не дуже важливо (2)
- ☐ взагалі не важливо (1)
- ☐ важко відповісти (0)

24. Оцініть важливість, яку має для Вас під час вибору програмних засобів віртуалізації наступний фактор: *

можливість керування програмним засобом із командного рядка

- ☐ дуже важливо (5)
- ☐ важливо (4)
- ☐ не обов'язково, але доречно (3)
- ☐ не дуже важливо (2)
- ☐ взагалі не важливо (1)
- ☐ важко відповісти (0)

25. Оцініть важливість, яку має для Вас під час вибору програмних засобів віртуалізації наступний фактор: *

наявність у програмного засобу україномовного інтерфейсу

- ☐ дуже важливо (5)
- ☐ важливо (4)
- ☐ не обов'язково, але доречно (3)
- ☐ не дуже важливо (2)
- ☐ взагалі не важливо (1)
- ☐ важко відповісти (0)

26. Оцініть важливість, яку має для Вас під час вибору програмних засобів віртуалізації наступний фактор: *

можливість віддаленого доступу до віртуалізованої ОС

- ☐ дуже важливо (5)
- ☐ важливо (4)
- ☐ не обов'язково, але доречно (3)
- ☐ не дуже важливо (2)
- ☐ взагалі не важливо (1)
- ☐ важко відповісти (0)

27. Оцініть важливість, яку має для Вас під час вибору програмних засобів віртуалізації наступний фактор: *

належність програмного засобу до безкоштовного, умовно безкоштовного, вільно поширюваного ПЗ або наявність спеціальних доступних пропозицій для навчальних закладів

- ☐ дуже важливо (5)
- ☐ важливо (4)
- ☐ не обов'язково, але доречно (3)
- ☐ не дуже важливо (2)
- ☐ взагалі не важливо (1)
- ☐ важко відповісти (0)

28. Оцініть важливість, яку має для Вас під час вибору програмних засобів віртуалізації наступний фактор: *

популярність програмного засобу

- ☐ дуже важливо (5)
- ☐ важливо (4)
- ☐ не обов'язково, але доречно (3)
- ☐ не дуже важливо (2)
- ☐ взагалі не важливо (1)
- ☐ важко відповісти (0)

29. Оцініть важливість, яку має для Вас під час вибору програмних засобів віртуалізації наступний фактор: *

наявність різноманітних форм безкоштовної підтримки користувачів програмного засобу (користувачька та технічна документація, форуми, списки розсилки тощо)

- ☐ дуже важливо (5)
- ☐ важливо (4)
- ☐ не обов'язково, але доречно (3)
- ☐ не дуже важливо (2)
- ☐ взагалі не важливо (1)
- ☐ важко відповісти (0)

30. Оцініть важливість, яку має для Вас під час вибору програмних засобів віртуалізації наступний фактор: *

дата випуску останнього стабільного релізу (актуальність програмного засобу)

- ☐ дуже важливо (5)
- ☐ важливо (4)
- ☐ не обов'язково, але доречно (3)
- ☐ не дуже важливо (2)
- ☐ взагалі не важливо (1)
- ☐ важко відповісти (0)

31. Можливо, Ви вважаєте важливими й інші фактори вибору програмних засобів віртуалізації, відсутні у цьому опитуванні. Якщо так - вкажіть їх, будь ласка, у наступному полі, зазначивши для кожного його важливість особисто для Вас.

32. Чи використовуєте Ви програмні засоби віртуалізації ОС під час лекцій?

- ☐ так
- ☐ ні

33. Оцініть співвідношення використання Вашими студентами графічного Інтерфейсу та командного рядку під час виконання лабораторних робіт:

- ☐ лише графічний інтерфейс
- ☐ лише командний рядок
- ☐ обидва варіанти, але з переважанням графічного інтерфейсу
- ☐ обидва варіанти, але з переважанням командного рядка
- ☐ обидва варіанти приблизно однаковою мірою

34. Оцініть співвідношення використання Вашими студентами графічного Інтерфейсу та командного рядка під час виконання лабораторних робіт, пов'язаних з моніторингом процесів та керуванням ними:

- ☐ лише графічний інтерфейс
- ☐ лише командний рядок
- ☐ обидва варіанти, але із переважанням графічного інтерфейсу
- ☐ обидва варіанти, але із переважанням командного рядка
- ☐ обидва варіанти приблизно однаковою мірою
- ☐ питання не розглядається на лабораторних роботах

35. Оцініть співвідношення використання Вашими студентами графічного Інтерфейсу та командного рядка під час виконання лабораторних робіт, пов'язаних з керуванням користувачами та групами:

- ☐ лише графічний інтерфейс
- ☐ лише командний рядок
- ☐ обидва варіанти, але із переважанням графічного інтерфейсу
- ☐ обидва варіанти, але із переважанням командного рядка
- ☐ обидва варіанти приблизно однаковою мірою
- ☐ питання не розглядається на лабораторних роботах

36. Оцініть співвідношення використання Вашими студентами графічного Інтерфейсу та командного рядка під час виконання лабораторних робіт, пов'язаних з маніпулюванням файлами:

- ☐ лише графічний інтерфейс
- ☐ лише командний рядок
- ☐ обидва варіанти, але із переважанням графічного інтерфейсу
- ☐ обидва варіанти, але із переважанням командного рядка
- ☐ обидва варіанти приблизно однаковою мірою
- ☐ питання не розглядається на лабораторних роботах

37. Оцініть співвідношення використання Вашими студентами графічного інтерфейсу та командного рядка під час виконання лабораторних робіт, присвячених роботі з пристроями:

- ☐ лише графічний інтерфейс
- ☐ лише командний рядок
- ☐ обидва варіанти, але із переважанням графічного інтерфейсу
- ☐ обидва варіанти, але із переважанням командного рядка
- ☐ обидва варіанти приблизно однаковою мірою
- ☐ питання не розглядається на лабораторних роботах

38. Оцініть співвідношення використання Вашими студентами графічного інтерфейсу та командного рядка під час виконання лабораторних робіт, пов'язаних з моніторингом використання пам'яті та керуванням нею:

- ☐ лише графічний інтерфейс
- ☐ лише командний рядок
- ☐ обидва варіанти, але із переважанням графічного інтерфейсу
- ☐ обидва варіанти, але із переважанням командного рядка
- ☐ обидва варіанти приблизно однаковою мірою
- ☐ питання не розглядається на лабораторних роботах

39. Оцініть співвідношення використання Вашими студентами графічного інтерфейсу та командного рядка під час виконання лабораторних робіт, пов'язаних зі встановленням, оновленням та деінсталяцією програмного забезпечення:

- ☐ лише графічний інтерфейс
- ☐ лише командний рядок
- ☐ обидва варіанти, але із переважанням графічного інтерфейсу
- ☐ обидва варіанти, але із переважанням командного рядка
- ☐ обидва варіанти приблизно однаковою мірою
- ☐ питання не розглядається на лабораторних роботах

40. Оцініть співвідношення використання Вашими студентами графічного інтерфейсу та командного рядка під час виконання лабораторних робіт, присвячених основам роботи з операційною системою та її базовими налаштуваннями (мережні з'єднання, клавіатура, миша, зовнішній вигляд тощо):

- ☐ лише графічний інтерфейс
- ☐ лише командний рядок
- ☐ обидва варіанти, але із переважанням графічного інтерфейсу
- ☐ обидва варіанти, але із переважанням командного рядка і/або редагування конфігураційних файлів
- ☐ обидва варіанти приблизно однаковою мірою
- ☐ питання не розглядається на лабораторних роботах

41. Чи наявний у Unix-подібній ОС, котру вивчають Ваші студенти, режим sudo, і чи передбачене його використання під час виконання лабораторних робіт?

- ☐ режим sudo наявний, передбачене його використання під час виконання лабораторних робіт
- ☐ режим sudo наявний, але під час виконання лабораторних робіт передбачений вхід від імені користувача root в обхід механізму sudo
- ☐ режим sudo відсутній, тож його використання під час виконання лабораторних робіт не передбачене
- ☐ Інше:

42. Які засоби наочності Ви використовуєте під час навчання ОС?

- ☐ електронна презентація
- ☐ трансляція через комп'ютерну мережу
- ☐ мультимедійний проектор
- ☐ дошка (звичайна, маркерна, фліпчарт тощо)
- ☐ мультимедійна дошка
- ☐ роздаткові матеріали
- ☐ друковані видання (наприклад, друковані навчально-методичні посібники)
- ☐ електронні видання (наприклад, електронні навчально-методичні посібники)
- ☐ Інше:

43. Які з перелічених нижче засобів навчання Ви застосовуєте під час навчання ОС?

- ☐ окремий веб-ресурс, присвячений навчальному предмету
- ☐ аудіозаписи лекцій
- ☐ відеозаписи лекцій
- ☐ електронні презентації
- ☐ тези лекцій в електронному форматі
- ☐ інструкції до лабораторних робіт в електронному форматі
- ☐ хмарні сервіси
- ☐ форум, присвячений навчальному предмету (чи відповідна тема в межах іншого форуму)
- ☐ вебіари
- ☐ електронна пошта
- ☐ соціальні мережі
- ☐ програми миттєвого обміну повідомленнями і/або відеозв'язку (Skype, Google Hangouts, Viber тощо)
- ☐ списки розсилки
- ☐ Інше:

44. Які з поданих нижче засобів навчання доступні для вільного завантаження Вашими студентами, що вивчають ОС?

- ☐ аудіозаписи лекцій
- ☐ відеозаписи лекцій
- ☐ електронні презентації
- ☐ тези лекцій в електронному форматі
- ☐ інструкції до лабораторних робіт в електронному форматі
- ☐ Інше:

45. Як Ви вважаєте, чи потрібні викладачу "Операційних систем та системного програмування" та інших подібних дисциплін спеціальні засоби підтримки, які б надавали відомості про наявні технології віртуалізації, що можуть бути застосовані для навчання ОС, та допомагали вибрати серед них таку, яка б найкраще підходила в кожному окремому випадку (довідкові матеріали, інфографіка, експертні системи тощо)? *

- ☐ так, потрібні
- ☐ так, було б не зайве, але нагальної потреби не бачу
- ☐ ні, не потрібні
- ☐ Інше:

46. Можливо, у Вас є досвід застосування технологій віртуалізації у навчанні ОС, яким би Ви хотіли поділитися.

Якщо це так, будь ласка, напишіть про це у наступному полі разом зі зручним для Вас форматом спілкування, координатами чи просто посиланням на відповідні публікації, якщо такі є. Спасибі!



Дякуємо, що знайшли час заповнити цю анкету!

Будемо раді Вашим зауваженням та пропозиціям за адресою olenagolovnya@gmail.com.

Надіслати

Ніколи не вказуйте паролі в Google Формах.

Додаток С.

ОПРАЦЮВАННЯ РЕЗУЛЬТАТІВ ФОРМУВАЛЬНОГО ЕТАПУ ЕКСПЕРИМЕНТУ ЗАСОБАМИ МОВИ R

Перевірка однорідності вибірки*Досвід роботи з Unix-подібними ОС (зокрема ОС Linux)*Програмний код

```
## Проведення тесту Фішера для результатів тестування
## однорідності вибірки (досвід роботи з Linux)
```

```
library("dplyr")
```

```
matrix_linux <-
  matrix(c(6, 44, 5, 48),
        nrow = 2,
        dimnames = list(EffectObserved = c("Yes", "No"),
                        Group = c("Experimental", "Control")))
fisher.test(matrix_linux, alternative = "two.sided")
```

Результати тесту

Fisher's Exact Test for Count Data

```
data: matrix_linux
p-value = 0.7561
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 0.3076023 5.8173919
sample estimates:
odds ratio
 1.305664
```

*Досвід роботи із засобами віртуалізації*Програмний код

```
## Проведення тесту Фішера для результатів тестування
## однорідності вибірки (досвід роботи із засобами віртуалізації)
```

```
library("dplyr")
```

```
matrix_vm <-
  matrix(c(4, 46, 6, 47),
        nrow = 2,
        dimnames = list(EffectObserved = c("Yes", "No"),
                        Group = c("Experimental", "Control")))
fisher.test(matrix_vm, alternative = "two.sided")
```

Результати тесту

Fisher's Exact Test for Count Data

```
data: matrix_vm
p-value = 0.7423
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 0.1328636 3.1005168
sample estimates:
odds ratio
0.6836807
```

Перевірка ефективності методики*Тестове опитування «Unix-подібні операційні системи»*Програмний код

```
## Проведення тесту Фішера для результатів
## тестового опитування "Unix-подібні операційні системи"
```

```
library("dplyr")
```

```
matrix_test_Unix <-
  matrix(c(32, 18, 19, 34),
        nrow = 2,
        dimnames = list(EffectObserved = c("Yes", "No"),
                        Group = c("Experimental", "Control")))
fisher.test(matrix_test_Unix, alternative = "two.sided")
```

Результати тесту

Fisher's Exact Test for Count Data

```
data: matrix_test_Unix
p-value = 0.005766
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 1.322439 7.711034
sample estimates:
odds ratio
3.143687
```

*Середній бал за захист лабораторних робіт*Програмний код

```
## Проведення тесту Фішера для середнього балу
## за захист лабораторних робіт
```

```
library("dplyr")

matrix_lab_assignments <-
  matrix(c(33, 17, 19, 34),
        nrow = 2,
        dimnames = list(EffectObserved = c("Yes", "No"),
                        Group = c("Experimental", "Control")))
fisher.test(matrix_lab_assignments, alternative = "two.sided")
```

Результати тесту

Fisher's Exact Test for Count Data

```
data: matrix_lab_assignments
p-value = 0.003051
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 1.435398 8.478342
sample estimates:
odds ratio
 3.429061
```

Перше контрольне тестове опитування з теорії операційних систем

Програмний код

```
## Проведення тесту Фішера для результатів першого контрольного
## тестового опитування з теорії операційних систем
```

```
library("dplyr")
matrix_control_test_1 <-
  matrix(c(30, 20, 17, 36),
        nrow = 2,
        dimnames = list(EffectObserved = c("Yes", "No"),
                        Group = c("Experimental", "Control")))
fisher.test(matrix_control_test_1, alternative = "two.sided")
```

Результати тесту

Fisher's Exact Test for Count Data

```
data: matrix_control_test_1
p-value = 0.005705
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 1.31652 7.72506
sample estimates:
odds ratio
 3.138996
```

Друге контрольне тестове опитування з теорії операційних систем

Програмний код

```
## Проведення тесту Фішера для результатів другого контрольного
## тестового опитування з теорії операційних систем
```

```
library("dplyr")
```

```
matrix_control_test_2 <-
  matrix(c(29, 21, 15, 38),
        nrow = 2,
        dimnames = list(EffectObserved = c("Yes", "No"),
                        Group = c("Experimental", "Control")))
fisher.test(matrix_control_test_2, alternative = "two.sided")
```

Результати тесту

Fisher's Exact Test for Count Data

```
data: matrix_control_test_2
p-value = 0.002893
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 1.430947 8.644717
sample estimates:
odds ratio
 3.453334
```

Додаток Т.
КОНТРОЛЬНЕ ТЕСТОВЕ ОПИТУВАННЯ
З ТЕОРІЇ ОПЕРАЦІЙНИХ СИСТЕМ №1

Операційні системи та системне програмування. Тест №1

Контрольний тест містить запитання з тем "Основні відомості про операційні системи", "Процеси та потоки" і "Пам'ять".

Введіть Ваше прізвище та ім'я: *

Ваша відповідь

Введіть номер Вашої групи: *

Ваша відповідь

Виберіть правильний варіант відповіді

Набір адрес, до яких процес має доступ, називається...

- ☐ потоком виконання
- ☐ деревом процесів
- ☐ адресним простором

Переходами між станами "Виконання" та "Готовності" керує компонент ОС під назвою...

- ☐ спеціальний драйвер
- ☐ планувальник процесів
- ☐ ядро

Сукупність одного чи більше потоків виконання та пов'язаних з ними ресурсів називається...

- ☐ простором користувача
- ☐ багатозадачною системою

☐ процесом

Механізм, за допомогою якого процеси, що виконуються у режимі користувача, можуть отримувати доступ до функцій режиму ядра, має назву...

☐ багатозадачність

☐ системний виклик

☐ підкачування

Системи FMS та IBSYS є прикладом...

☐ систем пакетної обробки

☐ систем з режимом поділу часу

☐ розподілених систем

В ОС цієї архітектури всі базові функції реалізовані у ядрі. Йдеться про...

☐ багаторівневі ОС

☐ ОС на основі мікроядра

☐ монолітні ОС

ОС розподіляє між процесами час ЦП в основному...

☐ у просторі

☐ у часі

☐ у контексті

ОС Mach, Chorus, Sprite належать до...

☐ мережних систем

☐ розподілених систем

☐ систем з режимом поділу часу

Ця ознака належності компонента ОС до ядра може виконуватися не завжди:

☐ виконання компонентом найважливіших функцій ОС

☐ робота у привілейованому режимі

- ☐ постійне перебування в оперативній пам'яті

Реалізація потоків у просторі користувача передбачає, що...

- ☐ ядро знає про потоки і керує ними
- ☐ ядро керує своїми потоками й нічого не знає про потоки простору користувача
- ☐ ядро нічого не знає про потоки, воно керує однопотоковими процесами

Це реальна адреса, за якою у пам'яті зберігаються дані. Вона визначається на рівні мікросхеми пам'яті. Йдеться про...

- ☐ фізичну адреса
- ☐ віртуальну адреса
- ☐ логічну адреса

Фізична адреса, з якої починається адресний простір процесу, має назву...

- ☐ база
- ☐ межа
- ☐ сегмент

Ці діапазони адрес мають змінну довжину. Йдеться про...

- ☐ сегменти
- ☐ сторінки
- ☐ фрейми

Перетворення логічних адрес у фізичні здійснює...

- ☐ блок керування пам'яттю (MMU)
- ☐ лічильник команд
- ☐ процес

Із перелічених нижче видів пам'яті найвища швидкість доступу у...

- ☐ регістрів ЦП
- ☐ жорстких дисків
- ☐ основної пам'яті

Із поданих нижче видів пам'яті найбільший обсяг у...

- ☐ реєстрів ЦП
- ☐ кешу ЦП
- ☐ основної пам'яті

Для пришвидшення пошуку часто використовуваних сторінок застосовують...

- ☐ таблицю фреймів
- ☐ асоціативну пам'ять
- ☐ таблицю сторінок

Даний алгоритм заміщення сторінок використовує циклічний список сторінок, відсортований за часом їхнього завантаження в основну пам'ять. Ідеться про...

- ☐ алгоритм FIFO
- ☐ алгоритм "другий шанс"
- ☐ алгоритм "годинник"

Цей алгоритм заміщення сторінок передбачає вилучення сторінки, яку було завантажено в основну пам'ять першою. Описаний алгоритм має назву...

- ☐ алгоритм FIFO
- ☐ алгоритм NRU
- ☐ оптимальний алгоритм

Множина сторінок, яка при підкачуванні тримається в основній пам'яті, називається...

- ☐ таблиця сторінок
- ☐ кеш трансляції
- ☐ резидентна множина

Явище, за якого деякий невеликий блок пам'яті (використовувався одним процесом) вже звільнено, а наступний за ним блок – ще ні (використовується іншим процесом), має назву...

- ☐ внутрішня фрагментація пам'яті
- ☐ зовнішня фрагментація пам'яті
- ☐ сегментація пам'яті

Це явище супроводжується появою у пам'яті великої кількості коротких несуміжних блоків, внаслідок чого система не в змозі виділити довгий блок, хоча загальний обсяг вільної пам'яті для цього достатній. Ідеться про...

- ☐ зовнішню фрагментацію пам'яті
- ☐ фрагментацію пам'яті
- ☐ сегментацію пам'яті

Виберіть усі правильні варіанти відповіді (один або декілька)

До системного програмного забезпечення належить (належать)...

- ☐ драйвери та утиліти
- ☐ операційна система загалом
- ☐ графічний редактор

Додаток У.
**КОНТРОЛЬНЕ ТЕСТОВЕ ОПИТУВАННЯ
З ТЕОРІЇ ОПЕРАЦІЙНИХ СИСТЕМ №2**

Операційні системи та системне програмування. Тест №2

Контрольний тест містить запитання з тем "Файлові системи", "Безпека", "Планування" і "Міжпроцесова та міжпоточкова взаємодія".

Введіть Ваше прізвище та ім'я: *

Ваша відповідь

Введіть номер Вашої групи: *

Ваша відповідь

Виберіть правильний варіант відповіді

Впорядкований набір даних, що зберігається у комп'ютерній системі під спільним ім'ям, називається...

- ☐ розділом
- ☐ атрибутом
- ☐ файлом

Властивості, притаманні файлам, мають й іншу загальноприйнятну назву - ...

- ☐ повноваження
- ☐ характеристики
- ☐ атрибути

Ім'я файлу, що складається з повного шляху до цього файлу, називається...

- ☐ абсолютне ім'я

- ☐ робоче ім'я
- ☐ відносне ім'я

Особливий файл, що містить відомості про деякий набір інших файлів і таким чином групує їх, називається...

- ☐ суперфайлом
- ☐ каталогом
- ☐ профайлом

Розділ жорсткого диску, куди встановлюється й звідки завантажується ОС, має назву...

- ☐ MBR
- ☐ розширений розділ
- ☐ первинний розділ

Індексний вказівник у таблиці FAT відповідає...

- ☐ розділу
- ☐ файлу
- ☐ кластеру

У файловій системі NTFS перші 16 записів таблиці MFT називаються...

- ☐ квотами
- ☐ метафайлами
- ☐ кластерами

Об'єкт, що зберігає атрибути файлу та адреси фрагментів цього файлу, у файловій системі UFS називається...

- ☐ кластер
- ☐ і-вузол
- ☐ метафайл

UID - це...

- ☐ ідентифікатор користувача у Windows

- ☐ ідентифікатор процесу у Windows
- ☐ ідентифікатор користувача у Unix-подібних ОС

І користувачі, і групи у Windows використовують у ролі свого ідентифікатора...

- ☐ UID
- ☐ SID
- ☐ GID

Процедура, в ході якої система перевіряє, чи має суб'єкт, що претендує на ресурс, право на використання цього ресурсу, називається...

- ☐ авторизацією
- ☐ аутентифікацією
- ☐ ідентифікацією

Unix-подібні ОС зберігають у файлі /etc/passwd...

- ☐ відомості про групи облікових записів
- ☐ відомості про облікові записи, окрім паролів
- ☐ лише паролі до облікових записів

У Windows входом користувачів у систему займається компонент підсистеми безпеки під назвою...

- ☐ winlogon
- ☐ getty
- ☐ SRM

Згідно з цим підходом до авторизації, для кожного ресурсу задається список суб'єктів, яким дозволено використовувати цей ресурс. Ідеться про...

- ☐ переліки можливостей (C-lists)
- ☐ принцип мінімальних повноважень
- ☐ списки керування доступом (ACL)

Вибором процесу, що має виконуватися наступним, займається спеціальний компонент ОС - ...

- ☐ програма інтерфейсу користувача
- ☐ планувальник
- ☐ пристрій керування пам'яттю

У системах цієї категорії дуже важливо уникнути тривалого "захоплення" ЦП якимсь одним процесом і, як наслідок, повільної реакції такої системи на запити. Тому планування у таких системах здійснюється передусім на основі алгоритмів з витісненням. Ідеться про...

- ☐ інтерактивні системи
- ☐ системи пакетної обробки
- ☐ системи реального часу

Маємо два потоки у linux. Пріоритет першого потоку дорівнює 120, пріоритет другого потоку становить 90. Тобто вищий пріоритет має...

- ☐ перший потік
- ☐ другий потік

Критична область - це...

- ☐ деякий спільно використовуваний ресурс
- ☐ область програми, що містить помилку, яку поки не було локалізовано
- ☐ область програми, яка передбачає звернення до спільно використовуваного ресурсу

Кажуть, що має місце ситуація взаємного блокування (deadlock), якщо...

- ☐ потік P1 не може отримати доступ до ресурсу P1, а тому не може виконуватися далі, щоб звільнити зайнятий ним ресурс P2; потік P2 не може отримати доступ до ресурсу P2, а тому не може виконуватися далі, щоб звільнити зайнятий ним ресурс P1
- ☐ потік P1 отримав доступ до ресурсу P1, після чого зазнав краху, не встигнувши звільнити ресурс P1, і тепер жодний інший потік не може одержати доступ цього ресурсу
- ☐ потік P1 першим отримав доступ до глобальної змінної X і зчитав її значення (100); потім доступ до змінної X отримав потік P2, який теж зчитав її значення (знову 100), а також збільшив його на 10 (стало 110); зрештою доступ до змінної X знову отримав потік P1 і збільшив її значення на 20, додавши 20 до попередньо зчитаного значення ($100 + 20 = 120$); у результаті маємо $X=120$, хоч повинні були б мати $X=100+10+20=130$

До алгоритмів планування з витісненням належить алгоритм...

- ☐ першим прийшов - першим обслужили (FIFO)
- ☐ спершу найкоротша задача (SJF)
- ☐ пріоритет найменшому часу до кінця виконання (SRT)

Виберіть усі правильні варіанти відповіді (один або декілька)

Таблиця розділів буває розташована у...

- ☐ BIOS
- ☐ GPT
- ☐ MBR

Журналювання підтримує файлова система (файлові системи)...

- ☐ FAT
- ☐ NTFS
- ☐ ext3

Ведення журналу аудиту є призначенням процесу (процесів)...

- ☐ syslogd
- ☐ winlogon
- ☐ eventlog

Додаток Ф.

ВІДОМОСТІ ПРО АПРОБАЦІЮ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЇ

Масові науково-практичні заходи міжнародного рівня

1. VII міжнародна науково-практична конференція «Інноваційні технології в освіті» (Україна, м. Ялта, Кримський гуманітарний університет, 20-22 вересня 2010 року). Форма участі – виступ із доповіддю на тему «Вільне програмне забезпечення у вивченні операційної системи Linux».

2. FOSS Lviv-2011 (Україна, м. Львів, Львівський національний університет імені Івана Франка, 1-4 лютого 2011 року). Форма участі – виступ з доповіддю на тему «Linux та VirtualBox у навчанні абстрактних понять теорії операційних систем».

3. Науково-практичний семінар «Міжнародна співпраця у просторі відкритої освіти» (Україна, м. Київ, Інститут інформаційних технологій і засобів навчання НАПН України, 5 листопада 2012 року). Форма участі – публікація тез на тему «Психолого-педагогічні особливості студентів напряму підготовки "Інформатика" в контексті навчання операційних систем».

4. IX міжнародна конференції ITEA-2014 (Україна, м. Київ, 26 листопада 2014 року). Форма участі – публікація тез на тему «Критерії добору віртуалізаційного ПЗ у навчанні Unix-подібних операційних систем».

5. VII міжнародна науково-технічна конференція «Інформаційно-комп'ютерні технології – 2016» (Україна, м. Житомир, Житомирський державний технологічний університет, 22-23 квітня 2016 року). Форма участі – виступ із доповіддю на тему «Поєднання кількох засобів віртуалізації у навчанні бакалаврів інформатики дисципліни "Операційні системи та системне програмування"».

Масові науково-практичні заходи всеукраїнського рівня

1. Звітна наукова конференція Інституту інформаційних технологій і засобів навчання НАПН України (Україна, м. Київ, Інститут інформаційних технологій і засобів навчання НАПН України, 29 березня 2012 року). Форма

участі – виступ з доповіддю на тему *«Систематизація технологій віртуалізації»*.

2. II всеукраїнська науково-практична конференція молодих науковців *«Наукова молодь – 2014»* (Україна, м. Київ, Інститут інформаційних технологій і засобів навчання НАПН України, 11 грудня 2014 року). Форма участі – публікація тез на тему *«Критерії добору програмних засобів віртуалізації Unix-подібних операційних систем у підготовці бакалаврів інформатики»*.

3. Всеукраїнська науково-практична конференція з міжнародною участю *«Сучасні інформаційні технології в освіті та науці»* (Україна, м. Житомир, Житомирський державний університет імені Івана Франка, 10-11 листопада 2016 року). Форма участі – публікація тез на тему *«Аналіз досвіду навчання операційних систем у зарубіжних вищих освітніх закладах»*.

4. II всеукраїнська науково-практична конференція з міжнародною участю *«Сучасні інформаційні технології в освіті та науці»*, присвячена 10-й річниці функціонування Інтернет-порталу E-OLYMP (Україна, м. Житомир, Житомирський державний університет імені Івана Франка, 9-10 листопада 2017 року). Форма участі – публікація тез на тему *«Розробка методичних рекомендацій із застосування технологій віртуалізації у курсі з операційних систем»*.

5. III всеукраїнська науково-практична конференція з міжнародною участю *«Сучасні інформаційні технології в освіті та науці»*, (Україна, м. Житомир, Житомирський державний університет імені Івана Франка, 8-9 листопада 2018 року). Форма участі – публікація тез на тему *«Хмарні інтегровані середовища розробки у курсі з операційних систем для студентів педагогічних спеціальностей: досвід і перспективи»*.

Додаток Х.

ВІДОМОСТІ ПРО ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЙНОГО ДОСЛІДЖЕННЯ

Міністерство освіти і науки України
Житомирський державний університет
імені Івана Франка
Вул. В. Бердичівська, 40,
м. Житомир, 10008
телефон /факс (0412) 43-14-17
E-mail: zu@zu.edu.ua Web: www.zu.edu.ua
код ЄДРПОУ 02125208



Ministry of Education and Science of Ukraine
Zhytomyr Ivan Franko State University
40, Velyka Berdychivska Str.,
City of Zhytomyr Ukraine, 10008
Tel/Fax (0412) 43-14-17
E-mail: zu@zu.edu.ua Web: www.zu.edu.ua
USREOU 02125208

Від 19.06.17 № 102
На № _____ від _____

Довідка

про впровадження результатів дисертаційного дослідження
Головні Олени Сергіївни
з теми "Методика застосування технологій віртуалізації Unix-подібних
операційних систем у підготовці бакалаврів інформатики"
у навчальний процес Житомирського державного університету
імені Івана Франка

Результати дисертаційного дослідження О. С. Головні використовувалися у навчальному процесі Житомирського державного університету імені Івана Франка у 2015-2016 н.р. Впровадження результатів наукового дослідження дисертантки здійснювалося у межах курсу "Операційні системи та системне програмування" для студентів другого курсу напряму підготовки "Інформатика".

Під час впровадження методики застосування технологій віртуалізації Unix-подібних операційних систем у підготовці бакалаврів інформатики використовувалися такі розробки О. С. Головні, як навчальна та робоча програма з дисципліни "Операційні системи та системне програмування", методичний посібник "Операційні системи та системне програмування", методичні рекомендації "Технології віртуалізації у навчанні операційних систем бакалаврів інформатики". Відзначався позитивний ефект від впровадження зазначених результатів, що полягав у підвищенні академічної успішності з дисципліни "Операційні системи та системне програмування" в студентів напряму підготовки "Інформатика", навчання яких здійснювалося відповідно до запропонованої О. С. Головнею методики. Спостерігалася помітна вмотивованість студентів до ґрунтовного виконання лабораторних робіт та їх своєчасного захисту.

Результати дисертаційної роботи О. С. Головні обговорено та схвалено на засіданні кафедри прикладної математики та інформатики (протокол №12 від 31 травня 2017 року).

Проректор з наукової та міжнародної роботи,
доктор педагогічних наук, професор

Завідувач кафедри прикладної математики
та інформатики,
кандидат педагогічних наук, доцент



Н. А. Сейко

Я. Б. Сікора

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДРОГОБИЦЬКИЙ ДЕРЖАВНИЙ ПЕДАГОГІЧНИЙ УНІВЕРСИТЕТ
імені ІВАНА ФРАНКА

вул. Івана Франка, 24, м. Дрогобич, 82100; тел. (0324) 41-04-74, факс (03244) 3-38-77
e-mail: administrator@drohobych.net, код ЄДРПОУ 02125438

14.05.2018 № 521

Довідка

про впровадження результатів дисертаційного дослідження
Головні Олени Сергіївни
на тему *«Методика застосування технологій віртуалізації Unix-подібних
операційних систем у підготовці бакалаврів інформатики»*

Протягом 2016-2017 років на базі навчально-наукового інституту фізики, математики, економіки та інноваційних технологій Дрогобицького державного педагогічного університету імені Івана Франка проводилось науково-педагогічне дослідження «Методика застосування технологій віртуалізації Unix-подібних операційних систем у підготовці бакалаврів інформатики» Головні О. С.

Здійснено апробацію методичного посібника «Операційні системи та системне програмування» та методичних рекомендацій «Технології віртуалізації у навчанні операційних систем бакалаврів інформатики» у межах навчальної дисципліни «Операційні системи та системне програмування» для бакалаврів інформатики. За підсумками експериментальної роботи виявлено інтенсифікацію навчального процесу, підвищення рівня мотивації студентів, поліпшення результатів їх навчання.

Аналіз запропонованої методики дає підстави зробити висновки про актуальність дослідження Головні О. С. та доцільність впровадження розробленої методики у процес підготовки бакалаврів інформатики.

Результати апробації запропонованої методики застосування технологій віртуалізації Unix-подібних операційних систем у підготовці бакалаврів інформатики було обговорено на засіданні кафедри інформатики та інформаційних систем (протокол №6 від 14 травня 2018).

Кандидат технічних наук,
доцент, завідувач кафедри
інформатики та інформаційних систем

Сікора О. В.

Сікора О. В.

Проректор з наукової роботи



Пантюк М. П.

Пантюк М. П.



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ ПЕДАГОГІЧНИЙ УНІВЕРСИТЕТ
імені А.С. МАКАРЕНКА

вул. Роменська, 87, м. Суми, 40002, факс (0542) 22-15-17, тел. (0542) 22-14-95
E-mail: rector@sspu.sumy.ua Код ЄДРПОУ 02125510

11.05.2018 № 1151 На № _____ від _____

ДОВІДКА

про впровадження результатів дисертаційного дослідження
Головні Олени Сергіївни
з теми *«Методика застосування технологій віртуалізації
unix-подібних операційних систем
у підготовці бакалаврів інформатики»*

Впродовж 2016-2017 рр. у Сумському державному педагогічному університеті імені А. С. Макаренка проводилося дослідження О. С. Головні з теми «Методика застосування технологій віртуалізації unix-подібних операційних систем у підготовці бакалаврів інформатики».

Результати дослідження О. С. Головні впроваджувалися у процес підготовки бакалаврів інформатики, зокрема у межах спецкурсу «Інформаційні системи» використовувалися методичний посібник «Операційні системи та системне програмування» і методичні рекомендації «Технології віртуалізації у навчанні операційних систем бакалаврів інформатики». Під час застосування запропонованої О. С. Головною методики зафіксовано покращення результатів навчання студентів, підвищення інтересу студентів до дисципліни.

Результати апробації розробленої О. С. Головною методики застосування технологій віртуалізації unix-подібних операційних систем у підготовці бакалаврів інформатики обговорено та схвалено на засіданні кафедри інформатики (протокол № 9 від 24 квітня 2018 року).

Ректор
СумДПУ імені А.С. Макаренка,
доктор педагогічних наук,
професор



Ю.О. Лянной

Завідувач кафедри інформатики,
доктор педагогічних наук, доцент

О. В. Семеніхіна



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

БЕРДЯНСЬКИЙ ДЕРЖАВНИЙ ПЕДАГОГІЧНИЙ УНІВЕРСИТЕТ

вул. Шмідта, 4, м. Бердянськ, Запорізька обл. 71100
E-mail: rector@bdpu.org.ua; http://bdpu.org

Тел. +38(06153) 3-62-44, факс +38(06153) 4-74-68
Код ЄДРПОУ 02125220

19.03.2019 № 57-39/247

На № _____ від _____

Довідка

про впровадження результатів дисертаційного дослідження
Головні Олени Сергіївни
з теми «Методика застосування технологій віртуалізації Unix-подібних операційних систем у підготовці бакалаврів інформатики»

У 2016-2017 рр. відбувалося впровадження у навчальний процес Бердянського державного педагогічного університету результатів дисертаційного дослідження Головні О. С. «Методика застосування технологій віртуалізації Unix-подібних операційних систем у підготовці бакалаврів інформатики».

Було здійснено апробацію методичного посібника «Операційні системи та системне програмування» і методичних рекомендацій «Технології віртуалізації у навчанні операційних систем бакалаврів інформатики» у межах навчальної дисципліни «Сучасні операційні системи». Спостерігалось покращення результатів навчання студентів, підвищення активності студентів під час лабораторних занять.

Результати апробації методики застосування технологій віртуалізації Unix-подібних операційних систем у підготовці бакалаврів інформатики, запропонованої Головною О. С., обговорено та схвалено на засіданні кафедри комп'ютерних технологій в управлінні та навчанні й інформатики Бердянського державного педагогічного університету (протокол № 1 від 30 серпня 2018 р.).

Проректор з науково-педагогічної роботи
Бердянського державного педагогічного
університету

В.М. Ліпич

Завідувач кафедри комп'ютерних технологій в
управлінні та навчанні й інформатики
Бердянського державного педагогічного
університету

В.Г. Хоменко

