DOI: https://doi.org/10.31392/NZ-udu-168-1.2026.02
УДК 378.147:004.451

Доровці Адам Федорович,
доктор філософії, доцент, доцент кафедри математики та інформатики
Закарпатський угорський університет імені Ференца Ракоці II
https://orcid.org/0000-0003-4038-4945
e-mail: daroci.adam@kmf.org.ua

Головач Йожеф Ігнацович,
професор, доктор технічних наук,
професор кафедри математики та інформатики
Закарпатський угорський університет імені Ференца Ракоці II
https://orcid.org/0000-0002-2297-8961
e-mail: holovacs.jozsef@kmf.org.ua

Кучінка Каталін Йожефівна,
доцент, кандидат фізико-математичних наук,
доцент кафедри математики та інформатики
Закарпатський угорський університет імені Ференца Ракоці I,
https://orcid.org/0000-0001-8129-9937
e-mail: kucsinka.katalin@kmf.org.ua

ПРАКТИКО-ОРІЄНТОВАНЕ НАВЧАННЯ ОПЕРАЦІЙНИХ СИСТЕМ У ВІРТУАЛІЗОВАНОМУ КРОСПЛАТФОРМНОМУ СЕРЕДОВИЩІ

У статті теоретично обґрунтовано дидактичну цілісність практико-орієнтованого навчання операційних систем у віртуалізованому кросплатформному середовищі. Актуальність теми зумовлена подвійним статусом операційної системи: вона є об'єктом засвоєння системних понять і водночас середовищем виконання професійно значущих дій. Мета дослідження полягає у розкритті зв'язків між віртуалізацією, роботою у Windows і Linux, графічним та командним інтерфейсами, спостереженням системного стану й сценарною автоматизацією. Застосовано теоретичний аналіз і синтез праць із методики навчання операційних систем, порівняння форм представлення спільних системних понять у двох платформах, систематизацію видів діяльності та структурно-функціональне тлумачення віртуального середовища. Предметне поле окреслено тематикою інсталяції й конфігурування ОС, облікових записів, файлових систем, мережевих параметрів і служб, командної оболонки, дозволів, процесів та Bash-сценаріїв. Встановлено, що віртуальна машина виконує не тільки інфраструктурну, а й пізнавальну функцію, перетворюючи поняття ресурсу, ізоляції та системного стану на спостережувані параметри й наслідки дії. Кросплатформне зіставлення виявляє інваріанти ідентичності, доступу, процесу, служби, файлової системи та мережевого інтерфейсу, тоді як перехід від графічної дії до команди і сценарію формалізує операцію та робить її відтворюваною. Наукова новизна полягає у визначенні п'яти взаємопов'язаних переходів: від фізичного комп'ютера до віртуалізованого ресурсу, від графічної дії до команди, від команди до спостережуваного стану, від платформної реалізації до системного інваріанта, від ручної операції до автоматизації. Зроблено висновок, що єдність зовнішньої практичної діяльності та внутрішнього понятєвого пояснення формує цілісне розуміння ОС як системи керування ресурсами, середовища виконання програм і об'єкта відповідальної професійної діяльності.

Ключові слова: *операційні системи; віртуалізація; практико-орієнтоване навчання; кросплатформне середовище; командний рядок; системне адміністрування; Bash-автоматизація, STEM-освіта.*

Операційна система у вищій IT-освіті постає одночасно як теоретичний об'єкт, програмне середовище та інструмент професійної діяльності. Її вивчення охоплює абстракції процесу, пам'яті, файлової системи, введення-виведення і захисту, але ці абстракції набувають операційного змісту лише у взаємодії з реальною системою. У CS2023 до предметної галузі «Операційні системи» включено системні принципи, процеси, паралельність, планування, керування пам'яттю, пристрої, файлові системи, захист і віртуалізацію [3, с. 205–216]. Відповідно, педагогічна проблема полягає у поєднанні поняттєвого пояснення з діяльністю, під час якої стан системи можна змінювати, спостерігати й пов'язувати з її внутрішніми механізмами.

Практико-орієнтований аспект курсу «Операційні системи» охоплює інсталяцію і конфігурування Windows та Linux, облікові записи, файлові системи, мережеві параметри і служби, графічний та командний інтерфейси, дозволи й автоматизацію в Bash. Ці складники утворюють єдине предметне поле, у якому ресурс, процес, ідентичність, доступ, служба та сценарій виявляються через спостережувані стани ОС.

У дослідженні Марк Р. Еблінг (M. R. Ebling) систематизовано два виміри навчання операційних систем: абстрактний або конкретний підхід і внутрішня або зовнішня перспектива [4, с. 3]. Внутрішня перспектива зосереджується на реалізації ядра, системних викликах, плануванні, пам'яті та файлових структурах. Її класичним втіленням є Nachos, що перетворює механізми ядра на програмовані об'єкти [2, с. 482]. Зовнішня перспектива розкриває системні принципи через встановлення, конфігурацію, спостереження, адміністрування та автоматизацію.

М. Бен-Арі (M. Ben-Ari) пов'язує засвоєння інформатики з побудовою ментальних моделей і підкреслює, що правильне виконання процедури саме по собі не засвідчує розуміння механізму [1, с. 259-260]. Практична діяльність набуває навчального значення, коли контекстно насичений конкретний досвід поєднується з критичною рефлексією, концептуалізацією та подальшим експериментуванням [7, с. 1064]. A. Gaspar та співавтори показали, що віртуалізація надала системним дисциплінам автентичне середовище з адміністративним контролем без втручання в інфраструктуру аудиторії [5, с. 131–132]. У літературі водночас переважає внутрішня конкретна перспектива [4, с. 11], тому теоретична цілісність зовнішньої кросплатформної діяльності залишається менш розробленою частиною проблеми.

Мета статті – теоретично обґрунтувати дидактичну цілісність практико-орієнтованого навчання операційних систем у віртуалізованому кросплатформному середовищі. Завдання дослідження полягають у визначенні освітньої функції віртуалізації, розкритті зв'язків між графічною, командною і програмованою формами взаємодії з ОС, систематизації

поняттєвих інваріантів Windows і Linux, характеристиці переходу від виконання системної дії до причинного пояснення її результату.

Робота має характер теоретичного педагогічного дослідження. Використано аналіз і синтез наукових положень, порівняння представлення системних понять у Windows і Linux, систематизацію діяльності та структурно-функціональне тлумачення віртуального середовища. Предметний контекст окреслено тематикою відкритого циклу лабораторних матеріалів [6]: віртуалізація й інсталяція ОС, адміністрування, мережеві служби, командний рядок, дозволи, процеси та Bash-сценарії. Матеріали визначили межі тематики без окремого аналізу їхніх завдань або структури.

Віртуальна машина відтворює зв'язок між апаратними ресурсами та програмними абстракціями в контрольованій формі. Виділення оперативної пам'яті, процесорних ядер, дискового простору і мережевого адаптера робить поняття ресурсу доступним ще до запуску гостьової системи. Подальші етапи інсталяції розкривають зв'язки між образом носія, завантаженням, таблицею розділів, файловою системою, точкою монтування, обліковим записом і системною конфігурацією. Отже, віртуалізація виконує не лише інфраструктурну, а й пізнавальну функцію: абстрактна архітектура комп'ютерної системи перетворюється на сукупність параметрів, станів і наслідків дії.

Ізоляція гостьової ОС від основної системи змінює характер навчальної помилки. Невдала конфігурація, помилково заданий параметр або некоректно встановлений пакет постають як спостережувана зміна стану, а не як незворотне пошкодження навчальної інфраструктури. У цьому середовищі інсталяція, перезапуск, перевірка і відновлення утворюють завершений життєвий цикл системної дії. Таким чином, можливість повторного виконання, перевірки й коригування системних дій створює умови для циклічного переходу від конкретного досвіду до рефлексії, концептуалізації та нового активного експериментування [7, с. 1065].

Windows і Linux на рівні інтерфейсу, термінології та службових засобів істотно відрізняються, однак їх паралельне вивчення виявляє спільні системні відношення. Обліковий запис у кожній системі пов'язує цифрову ідентичність із набором прав [8, с. 33–34], файлова система організує імена, каталоги, метадані та доступ [8, с. 529–550], процес є екземпляром програми, що виконується, разом із її поточним станом [8, с. 106]. Мережевий інтерфейс поєднує локальну конфігурацію із зовнішньою комунікацією [8, с. 56–57], служба підтримує тривалу фонову функцію [8, с. 74–75], а оболонка перетворює введені користувачем команди на операції над системними об'єктами [8, с. 58–59].

Кросплатформність у цьому контексті означає не механічне дублювання однакової процедури. Вона виявляє різні реалізації спільного принципу: графічне керування користувачами співвідноситься з командами створення облікових записів, списки контролю доступу Windows – із моделлю власника, групи та бітів дозволів Linux, диспетчер завдань – із командами спостереження

за процесами, параметри мережевого адаптера – з текстовим виведенням мережевої конфігурації. Порівняння переводить увагу з назви елемента інтерфейсу на функцію, яку він реалізує.

Графічний інтерфейс подає системні об'єкти через візуальну ієрархію, доступні дії та безпосередній зворотний зв'язок. Командний рядок репрезентує ті самі об'єкти через формальну мову: ім'я команди, параметри, потоки введення і виведення, код завершення. Перехід між цими формами взаємодії поглиблює розуміння, оскільки один системний стан отримує дві взаємодоповнювальні репрезентації. Візуальна дія показує місце об'єкта у структурі системи, а команда робить явними операцію, аргументи й результат.

Оболонка Linux додає до окремих команд композиційність. Конвеєр пов'язує стандартний вивід одного процесу зі стандартним введенням іншого, перенаправлення перетворює потік на файл, фільтрація виокремлює релевантні рядки, змінні й позиційні параметри відокремлюють алгоритм від конкретних даних. Bash-сценарій є точкою переходу від ручного адміністрування до програмованої системної діяльності. У ньому системні поняття взаємодіють з алгоритмічними конструкціями: послідовністю, розгалуженням, параметризацією, перевіркою умов і завершенням процесу.

Практична дія набуває навчального змісту тоді, коли її результат співвідноситься зі станом системи. Відомості про версію ядра, завантаження процесора, використання пам'яті й диска, список процесів, мережеві адреси, маршрути, активні з'єднання та права доступу утворюють систему спостережуваних ознак. Вони надають змогу розрізняти команду як текст, виконаний процес, зміну конфігурації та фактичний ефект. Причинне пояснення з'єднує ці рівні. Дія користувача змінює параметр або об'єкт, ядро і служби реалізують зміну, діагностичний засіб фіксує новий стан.

Така логіка долає розрив між декларативним знанням і процедурним умінням. Декларативний рівень відповідає на питання про призначення процесу, дозволу чи маршруту. Процедурний – відтворює операцію, а причинний – пояснює, чому отримано саме такий результат і які компоненти системи в ньому беруть участь. Науковий результат теоретичного узагальнення полягає у визначенні п'яти взаємопов'язаних переходів, що структурують практико-орієнтоване навчання ОС: від фізичного комп'ютера до віртуалізованого ресурсу, від графічної дії до формалізованої команди, від команди до спостережуваного стану, від платформної реалізації до системного інваріанта, від ручної операції до сценарної автоматизації.

Адміністративні повноваження надають практичній роботі автентичності, але водночас роблять безпеку внутрішньою складовою предмета. Розмежування звичайного й привілейованого користувача розкриває принцип мінімальних повноважень, дозволи демонструють зв'язок між ідентичністю та доступом, мережеві режими віртуальної машини показують межу ізоляції, журнали й діагностичні відомості фіксують наслідки системних змін.

Адміністративна дія виступає способом пізнання операційної системи ззовні, а концептуальне пояснення пов'язує її з внутрішніми механізмами. Ці

перспективи не конкурують: перша забезпечує спостережуваний досвід, друга надає йому понятєву структуру. Їх єдність формує уявлення про ОС як систему керування ресурсами, середовище виконання програм і об'єкт відповідальної професійної діяльності.

Висновки. Практико-орієнтоване навчання операційних систем у віртуалізованому кросплатформному середовищі має цілісну дидактичну основу. Віртуалізація операціоналізує поняття ресурсу, ізоляції та життєвого циклу системної зміни. Зіставлення Windows і Linux виявляє інваріанти ідентичності, доступу, файлової системи, процесу, служби та мережевого інтерфейсу. Взаємодія графічної оболонки, командного рядка і Bash-сценарію відображає послідовний перехід від візуальної операції до формалізації й автоматизації.

Наукова новизна дослідження полягає у теоретичному визначенні п'яти переходів, через які зовнішня діяльність із системою пов'язується з її понятєвим змістом. Перспективу подальших досліджень становить емпіричне вивчення того, як ці переходи впливають на здатність пояснювати причинні зв'язки та переносити системні поняття між Windows і Linux.

Використані джерела:

1. Ben-Ari M. Constructivism in computer science education. *ACM SIGCSE Bulletin*. 1998. Vol. 30. № 1. P. 257–261. DOI: https://doi.org/10.1145/274790.274308.
2. Christopher W. A., Procter S. J., Anderson T. E. The Nachos Instructional Operating System. *USENIX Winter 1993 Conference. San Diego, CA :* USENIX Association, 1993. P. 481–488. URL: https://www.usenix.org/conference/usenix-winter-1993-conference/nachos-instructional-operating-system.
3. Kumar A. N. et al. Computer Science Curricula 2023. New York : ACM Press, IEEE Computer Society Press, AAAI Press, 2024. 459 p. DOI: https://doi.org/10.1145/3664191.
4. Ebling M. R. Resources for Teaching Operating Systems: A Survey of Instructors and a Literature Review. *ACM Transactions on Computing Education*. 2024. Vol. 24. № 4. P. 1–28. DOI: https://doi.org/10.1145/3688853.
5. Gaspar A., Langevin S., Armitage W. D., Sekar R., Daniels T. E. The role of virtualization in computing education. *Proceedings of the 39th ACM Technical Symposium on Computer Science Education (SIGCSE '08).* New York : ACM Press, 2008. P. 131–132. DOI: https://doi.org/10.1145/1352135.1352181.
6. Il. RF KMF DokuWiki. Operating Systems: Laboratory Works 1–6. URL: https://okt.kmf.uz.ua/dw/doku.php?id=informatika3:laborok:labor1&do=index.
7. Morris T. H. Experiential learning – a systematic review and revision of Kolb's model. *Interactive Learning Environments.* 2020. Vol. 28. № 8. P. 1064–1077. DOI: https://doi.org/10.1080/10494820.2019.1570279.
8. Silberschatz A., Galvin P. B., Gagne G. Operating System Concepts. 10th ed. Hoboken, NJ : Wiley, 2018. 1040 p.

References:

1. Ben-Ari M. (1998) Constructivism in computer science education. *ACM SIGCSE Bulletin*. Vol. 30. № 1. P. 257–261. DOI: https://doi.org/10.1145/274790.274308 [in English].
2. Christopher W. A., Procter S. J., Anderson T. E. (1993) The Nachos Instructional Operating System. *USENIX Winter 1993 Conference. San Diego,* CA : USENIX Association. P. 481–488. URL: https://www.usenix.org/conference/usenix-winter-1993-conference/nachos-instructional-operating-system [in English].

3. Kumar A. N. et al. (2024) Computer Science Curricula 2023. New York : ACM Press, IEEE Computer Society Press, AAAI Press. 459 p. DOI: https://doi.org/10.1145/3664191 [in English].
4. Ebling M. R. (2024) Resources for Teaching Operating Systems: A Survey of Instructors and a Literature Review. *ACM Transactions on Computing Education*. Vol. 24. № 4. P. 1–28. DOI: https://doi.org/10.1145/3688853 [in English].
5. Gaspar A., Langevin S., Armitage W. D., Sekar R., Daniels T. E. (2008) The role of virtualization in computing education. *Proceedings of the 39th ACM Technical Symposium on Computer Science Education (SIGCSE '08)*. New York : ACM Press. P. 131–132. DOI: https://doi.org/10.1145/1352135.1352181 [in English].
6. II. RF KMF DokuWiki. (n.d.) Operating Systems: Laboratory Works 1–6. URL: https://okt.kmf.uz.ua/dw/doku.php?id=informatika3:laborok:labor1&do=index [in English].
7. Morris T. H. (2020) Experiential learning – a systematic review and revision of Kolb's model. Interactive Learning Environments. Vol. 28. № 8. P. 1064–1077. DOI: https://doi.org/10.1080/10494820.2019.1570279 [in English].
8. Silberschatz A., Galvin P. B., Gagne G. (2018) Operating System Concepts. 10th ed. Hoboken, NJ : Wiley. 1040 p. [in English].

A. Dorovtsi, J. Holovács, K. Kuchinka. Practice-oriented teaching of operating systems in a virtualized cross-platform environment.

The article substantiates the didactic coherence of practice-oriented operating systems education in a virtualized cross-platform environment. An operating system has a dual status: it is both an object for learning systems concepts and an environment for performing meaningful technical actions. The study aims to reveal the relationships among virtualization, work in Windows and Linux, graphical and command-line interfaces, observation of system state, and script-based automation. The methods comprise theoretical analysis and synthesis of research on operating systems education, comparison of how common concepts are represented on the two platforms, systematization of learning activities, and structural-functional interpretation of the virtual environment. The subject scope includes operating system installation and configuration, user accounts, file systems, network parameters and services, the command shell, permissions, processes, and Bash scripts. The findings show that a virtual machine has both infrastructural and epistemic functions: it turns resource, isolation, and system state into observable parameters and consequences of action. Cross-platform comparison reveals invariants of identity, access, process, service, file system, and network interface, while the transition from a graphical action to a command and then a script formalizes an operation and makes it reproducible. The scientific novelty lies in identifying five interrelated transitions: from a physical computer to a virtualized resource; from a graphical action to a command; from a command to an observable state; from a platform-specific implementation to a systems invariant; and from a manual operation to automation. The unity of external practical activity and internal conceptual explanation supports an integrated understanding of an operating system as a resource manager, a program execution environment, and an object of responsible professional activity.

Keywords: *operating systems; virtualization; practice-oriented learning; cross-platform environment; command line; system administration; Bash automation, STEM education.*

Дата першого надходження рукопису до видання: 06.05.2026

Дата прийнятого до друку рукопису після рецензування: 21.05.2026

Дата публікації: 26.06.2026