

tekhnologiy na osnovi adkhokratychnoho ta empauermentnoho pidkhodiv. Naukovyy visnyk NUBiP Ukrainy. Seriya: Pedagogika, psikhologiya, filosofiya. 2018. Vyp. № 279 (2018). S.74-84. URL: <http://journals.nubip.edu.ua/index.php/Pedagogica/article/view/12073>. (data zvernennya: 08.10.18.).

12. Klochko V. I., Bondarenko Z. V. Informatsiyno-komunikatsiyni tekhnolohiyi yak zasib formuvannya doslidnyts'kykh umin' studentiv tekhnichnykh universytetiv Visnyk Vinnyts'koho politekhnichnoho instytutu. 2009. Vyp. № 1. S. 102-106. URL: <https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/5916/707.pdf?sequence=3&isAllowed=y>. (data zvernennya: 08.10.18.).

Theoretical and methodological principles of the implementation of the adhocratic approach to study the disciplines of mathematical cycle using systems of computer mathematics

Klochko V.I., Klochko O.V.

Abstract. The article describes the goals, content and how to implement adhocratic approach during the study disciplines mathematical cycle using modern information and communication technologies using computer systems mathematics. An important task of modern higher education is the formation of the ability of students to apply information and communication technologies in future careers. This will facilitate the process of self-education, self-improvement, creative personal development throughout their lives. Adhocratic implementation approach ensures continuous personal development, process efficiency study mathematical cycle using computer systems in mathematics. This is confirmed by the results of pedagogical experiments using these and other professionally-oriented tasks.

Keywords: training of future specialists, adhocratic approach, information and communications technology, computer mathematics system, the discipline of mathematics, applied problems.

УДК 004.774:004.946.5

В.М. Франчук

кандидат педагогічних наук, професор
Національний педагогічний університет імені М.П. Драгоманова

Використання веб-орієнтованого віртуального середовища Proxmox в педагогічних закладах освіти

Анотація. В статті розглядаються засоби для створення віртуальних машин з використанням веб-орієнтованої системи Proxmox VE за допомогою трьох відкритих технологій віртуалізації, а саме: KVM, QEMU, LXC. Використання Proxmox VE у закладах освіти може мати такі переваги, як покращення рівня обслуговування комп'ютерного обладнання, спрощення управління і захисту даних, зниження вартості апаратних засобів та електроживлення, а також переваги віртуалізації під час розробки в середовищах тестування. Саме використання веб-орієнтованого інтерфейсу системи дозволяє здійснювати управління віртуальними машинами з використанням сучасних підходів у віртуалізації, тому у статті більш детально описано використання та створення віртуальних машин за допомогою відкритих технологій віртуалізації.

Ключові слова: віртуалізація, віртуальні машини, контейнеризація.

Широке розповсюдження різноманітного програмного забезпечення, за допомогою якого можна створювати та використовувати віртуальні машини (комп'ютери), а також використання хмарних сервісів, які останнім часом набули широкої популярності, потребують ретельного вивчення та детального аналізу досвіду впровадження сучасних інформаційно-комунікаційних технологій в навчальний процес. Окрім теоретичної підготовки важливою складовою підготовки майбутніх фахівців з інформаційних технологій та вчителів інформатики до майбутньої трудової діяльності, пов'язаної з проектуванням, конфігуруванням та управлінням функціонуванням мережі та мережевих сервісів, є організація лабораторних занять, адже саме практичні навички роботи з різноманітними операційними системами, налаштуванням мережевих сервісів та відповідного програмного забезпечення, є головною складовою у формуванні знань, умінь та навичок майбутніх фахівців з інформаційних технологій та вчителів інформатики [4].

Досвід впровадження хмарних сервісів та сучасних інформаційно-комунікаційних технологій в навчальний процес висвітлено у працях В.Ю. Бикова [1], М.І Жалдака [3], Н.М. Морзе, З.С. Сейдаметової та інших, впровадження в навчальний процес віртуальних машин (комп'ютерів) розглядається у роботах О.В. Наумука [4], В.П. Олексюка [5], С.М Яшанова [7] та інших.

Аналіз сучасних підходів до організації навчального процесу [4, 5, 7] дозволяє стверджувати, що впровадження віртуальних машин, реалізованих за допомогою програмних засобів, та різноманітних веб-орієнтованих сервісів дозволяє повністю задовольнити потреби для проведення практичних занять пов'язаних з проектуванням, конфігуруванням, управлінням функціонуванням мережевих сервісів. Крім того використання веб-орієнтованих сервісів надає ряд можливостей, таких як використання веб-

браузерів для доступу до віртуальних машин, що дозволяє використовувати ці віртуальні машини навіть за допомогою мобільних пристроїв, створювати стендові середовища для розробки, демонстрації та тестування програмного забезпечення, більш раціонально використовувати апаратні та програмні ресурси, що робить використання веб-орієнтованих сервісів перспективним для подальшого розвитку та потребує додаткового вивчення [4].

В цій статті розглянуто досвід впровадження спеціального програмного забезпечення, використання якого дозволяє створювати віртуальні машини для організації практичних та лабораторних занять з дисциплін, пов'язаних з налаштуванням серверних операційних систем та мережевих сервісів.

Використання віртуальних машин тісно пов'язане з поняттям віртуалізації. Віртуалізація – це процес надання набору обчислювальних ресурсів або їх логічне об'єднання, яке абстраговане від апаратної реалізації, за допомогою яких забезпечується логічна ізоляція один від одного обчислювальних процесів, які виконуються на певних фізичних ресурсах.

Прикладом використання віртуалізації є можливість запуску кількох операційних систем на одному комп'ютері. Для кожної з таких гостьових операційних систем виокремлюються відповідні набори логічних ресурсів (процесорних, оперативної пам'яті, пристроїв зберігання даних), які надаються із загального масиву, доступного на рівні обладнання, і управляються за допомогою базової (хостової) операційної системи, а саме програми – гіпервізора (англ. Hypervisor; від ст.-греч. ὑπέρ «над, вище» + лат. visio «зір; бачення», гіпервізор може бути 1 типу (автономний), 2 типу (вбудований в операційну систему) та типу 1+ (гібридний)) або монітор віртуальних машин. Також віртуалізація може бути застосована до мережі передавання даних (наприклад віртуальна приватна мережа (Virtual Private Network)), сховищ зберігання даних (наприклад, віртуальна файлова система (Virtual File System)), прикладного програмного забезпечення (наприклад, віртуалізація додатків, сервісів), пам'яті (віртуальна пам'ять), бази даних, обладнання (емуляція – повна віртуалізація, наприклад, QEMU).

Для віртуалізації операційних систем застосовується кілька підходів, які за типом реалізації поділяються на програмні та апаратні [2].

Програмна віртуалізація поділяється ще на кілька підтипів:

- *Динамічна (бінарна) трансляція* – процес віртуалізації, коли проблемні команди гостьової операційної системи перехоплюються за допомогою гіпервізора і замінюються на безпечні.
- *Паравіртуалізація* – процес віртуалізації, під час якого для функціонування гостьової операційної системи створюється віртуалізоване середовище, в якому ядро цієї системи незначно модифікується. Функціонування гостьової операційної системи забезпечується за допомогою програми гіпервізора з використанням API (англ. Application Programming Interface). Використання паравіртуалізації дозволяє досягнути більш високої продуктивності операційної системи, у порівнянні з динамічною трансляцією. Паравіртуалізацію можна застосовувати лише в тому випадку, коли в гостьових операційних системах є відкриті вихідні коди, які можна модифікувати відповідно до їх ліцензії, або ж гіпервізор та гостьова операційна система розроблені одним виробником з врахуванням можливості паравіртуалізації гостьової системи.
- *Вбудована віртуалізація* – це підхід, який базується на застосуванні апаратно підтримуваної віртуалізації, за допомогою чого користувачі можуть використовувати будь-які версії операційних систем в поєднанні з різними варіантами робочих середовищ. Іншими словами, вбудована віртуалізація є повною віртуалізацією, що реалізована на апаратному рівні.

До основних переваг програмної віртуалізації можна віднести:

- Спільне використання ресурсів кількома гостьовими операційними системами (каталоги, принтери та ін.).
- Зручність інтерфейсу для вікон додатків з різних систем (вікна додатків виглядають так само, як в базовій системі).
- Під час налаштування апаратної платформи продуктивність роботи мало відрізняється від оригінальної операційної системи. Швидке перемикання між системами.
- Проста процедура відновлення гостьової операційної системи.
- Двостороння віртуалізація (програми однієї системи можна запускати в іншій і навпаки).

Апаратна віртуалізація – це процес віртуалізації, на основі апаратної підтримки. В разі використання цього процесу немає принципових відмінностей від процесу програмної віртуалізації. За допомогою апаратної віртуалізації забезпечується продуктивність рівня віртуальної машини, завдяки чому цей процес віртуалізації отримав широке практичне застосування. Прикладом використання апаратної віртуалізації можуть бути такі платформи: IBM LPAR, VMware, Hyper-V, Xen, KVM та ін.

До основних переваг апаратної віртуалізації можна віднести:

- Спрощення розробки програмних платформ віртуалізації за рахунок надання апаратних

інтерфейсів управління і підтримки віртуальних гостьових систем, що дозволяє зменшити трудомісткість і час на розробку систем віртуалізації.

- Збільшення швидкодії платформ віртуалізації. Управління віртуальними гостьовими системами здійснюється безпосередньо за допомогою гіпервізора, використання якого дозволяє досягти збільшення швидкодії.
- Збільшення захищеності даних. З'являється можливість перемикання між кількома запущеними незалежними платформами віртуалізації на апаратному рівні. Використання кожної з віртуальних машин може виконуватися незалежно від інших, в своєму просторі апаратних ресурсів, повністю ізольованих один від одного. Це дозволяє запобігти втраті швидкодії на базовій платформі і збільшити захищеність даних.
- Гостьова система не є прив'язаною до архітектури базової платформи і до реалізації платформи віртуалізації. Використання апаратної віртуалізації дає можливість запускати 64-бітні гостьові системи на 32-бітних базових системах.

Взагалі віртуалізацію можна використовувати під час роботи з операційними системами, віртуальними машинами, ресурсами та програмами. Розглянемо детальніше використання віртуалізації на рівні операційної системи та з використанням віртуальних машин.

Використання віртуалізації на рівні операційної системи дозволяє запускати ізольовані і безпечні віртуальні машини на одному фізичному вузлі, але не дозволяє запускати операційні системи з ядрами, відмінними від типу ядра базової операційної системи. Під час віртуалізації на рівні операційної системи не існує окремої програми гіпервізора. Замість цього у базовій операційній системі відбувається розподіл апаратних ресурсів між кількома віртуальними машинами і підтримка їх незалежності один від одного з використанням засобів самої операційної системи. Такий підхід називають *контейнеризацією*. Прикладами реалізацій такого підходу можуть бути: iCore Virtual Accounts, FreeBSD Jail, FreeVPS, Linux-VServer, LXC (Linux Containers), OpenVZ, Solaris Containers/Zones, Virtuozzo та ін.

Використання віртуальної машини дозволяє створювати середовище для гостьової операційної системи, як апаратне. Однак насправді це програмне середовище створюється на основі програмного забезпечення базової системи. Це середовище повинне бути достатньо надійним, щоб програмне забезпечення гостьової системи могло стабільно функціонувати. Під час використання паравіртуалізації, для віртуальної машини не створюється апаратне забезпечення, а, натомість, пропонується використовувати спеціальний API. Прикладами реалізацій такого підходу можуть бути: Parallels Workstation, QEMU, Virtual PC, VirtualBox, VMware Player, VMware Workstation.

Серед такої кількості реалізацій та підходів щодо використання віртуальних машин для організації практичних та лабораторних занять з дисциплін, пов'язаних з налаштуванням серверних операційних систем та мережевих сервісів, принциповим критерієм стосовно вибору відповідної технології можуть бути спосіб та умови використання цієї технології. Як відомо, всі проекти, реалізовані з використанням Linux-подібних операційних систем, реалізуються з відкритим кодом на умовах ліцензії GPL (General Public License). Одним із таких проектів є Proxmox Virtual Environment (Proxmox VE або PVE).

Proxmox VE реалізується в рамках проекту операційної системи Debian та є гіпервізором 2-го типу (Рис. 1), оскільки рівень віртуалізації вбудований в операційну систему, за допомогою якої створюється, реалізується, налаштовується та здійснюється управління віртуальними машинами.

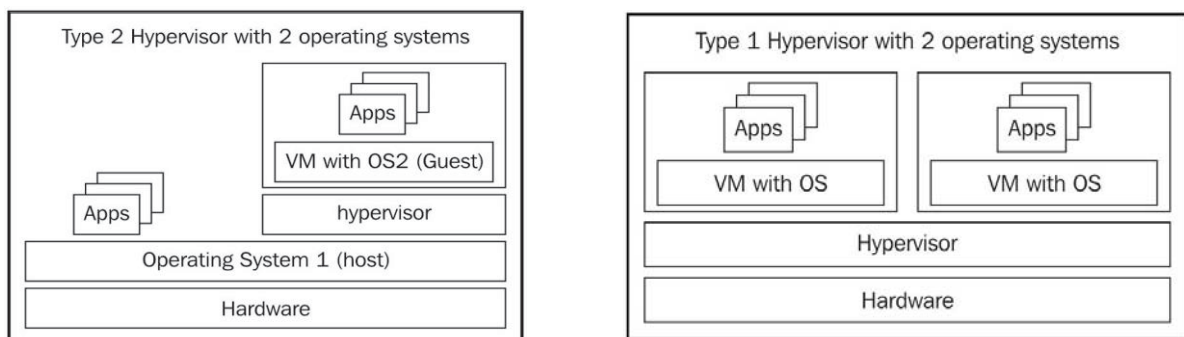


Рис. 1. Різниця між гіпервізором 1 та 2 типу

В Proxmox VE використовується три відкриті технології віртуалізації, а саме: KVM, QEMU, LXC – управління якими здійснюється через уніфікований веб-інтерфейс (Рис. 2).

Створювати та управляти віртуальними машинами можна за допомогою комбінації двох технологій з відкритим вихідним кодом: Kernel-based Virtual Machine (або KVM) і Quick Emulator (QEMU).

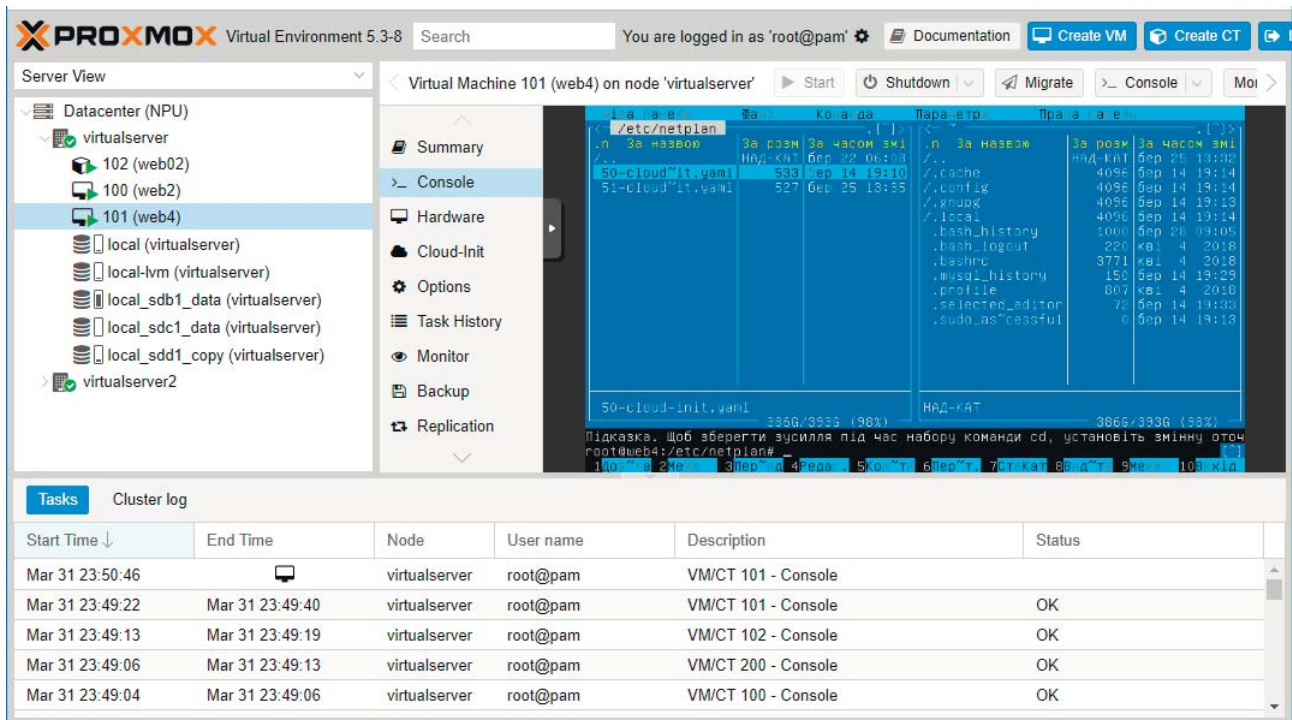


Рис. 2. Веб-інтерфейс управління Proxmox VE

Технологія KVM була додана як модуль ядра операційної системи Linux ще в лютому 2007 року. Використання цього модуля ядра дозволяє користувачам і адміністраторам операційної системи Linux отримувати переваги від апаратної архітектури віртуалізації.

Використання технології QEMU надає можливість створювати інтерфейс віртуалізації, яким можна управляти з командного рядка. За допомогою QEMU з командного рядка можна резервувати апаратні засоби, створити віртуальний диск і запускати чи зупиняти віртуальні машини. Проте здійснювати управління віртуалізацією за допомогою інструментів і функцій, які надаються у веб-інтерфейсі Proxmox VE – незважаючи на його залежність від QEMU (Рис. 3), користувачі не можуть. Управління віртуальними машинами з використання технології QEMU здійснюється за допомогою API.

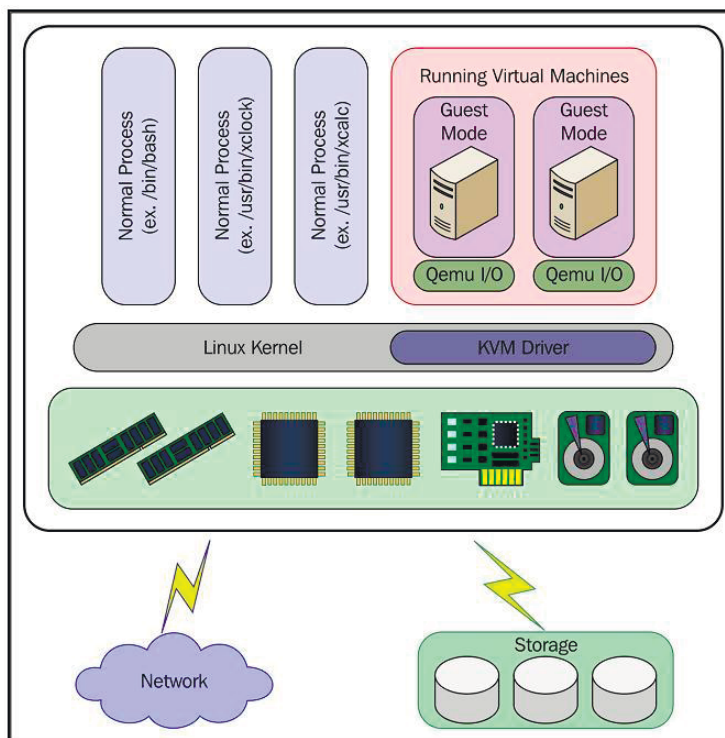


Рис. 3. Візуалізація взаємозв'язків між KVM та QEMU

Ще одна основна технологія, яка використовується в Proxmox VE, це технологія LXC (Linux Containers), починаючи з версії 4.0. До цієї версії використовувалася технологія OpenVZ. Використання

контейнерів є іншим типом віртуалізації. На противагу віртуальним машинам, за допомогою контейнерів спільно використовуються компоненти операційної системи, такі як бібліотеки і виконувані файли базової операційної системи, чого не може відбуватися під час використання віртуальних машин.

На Рис. 4 показано порівняння віртуальних машин і контейнерів. У зв'язку із цим під час вибору між рішенням віртуальної машини і рішенням контейнера слід враховувати деякі обставини. Оскільки за допомогою контейнерів у Proxmox VE спільно використовуються компоненти базової операційної системи, гостьова операційна система має бути повністю реалізована за допомогою системи Debian, яка є операційною системою гіпервізора.

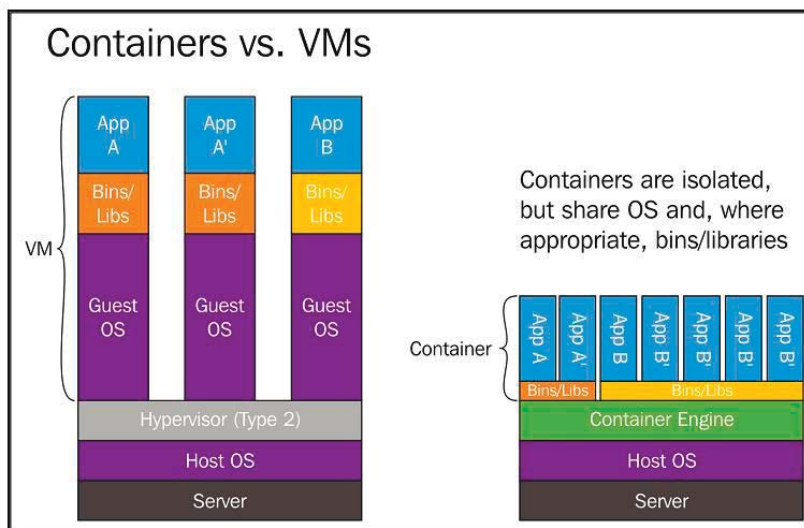


Рис. 4. Порівняння віртуальних машин і контейнерів

Коли потрібно, наприклад, використовувати операційну систему Microsoft Windows або іншу відмінну від Linux операційну систему, то використання рішення за допомогою контейнерів Proxmox VE перестає бути ефективним. В таких випадках замість використання контейнеризації слід створювати та використовувати віртуальні машини.

Отже за допомогою Proxmox VE забезпечується використання трьох технологій з відкритим вихідним кодом, а саме контейнеризацію і віртуалізацію за допомогою LXC, KVM і QEMU.

Слід зазначити, що використання Proxmox VE у закладах освіти може мати такі переваги, як покращення рівня обслуговування комп'ютерного обладнання, спрощення управління системою і захисту даних, зниження вартості апаратних засобів та електроживлення, а також переваги віртуалізації під час розробки в середовищах тестування. Крім цього під час використання віртуальних машин за допомогою сервісів міграції можна здійснювати міграцію цих віртуальних машин з одного сервера на інший, без зупинок і без перерв в обслуговуванні.

Список використаних джерел:

1. Биков В.Ю. Хмарна комп'ютерно-технологічна платформа відкритої освіти та відповідний розвиток організаційно-технологічної будови ІТ підрозділів навчальних закладів. *Теорія і практика управління соціальними системами: філософія, психологія, педагогіка, соціологія*. 2013. № 1. С. 81-98.
2. Віртуалізація. URL: <https://uk.wikipedia.org/wiki/Віртуалізація>. (дата звернення: 04.04.2019).
3. Жалдак М.І. Проблеми інформатизації навчального процесу в середніх і вищих навчальних закладах. *Комп'ютер у школі та сім'ї*. 2013. № 3. С. 8-15.
4. Научук О.В. Стан та перспективи впровадження засобів віртуалізації у процес вивчення дисципліни «Адміністрування комп'ютерних мереж» *Інформаційні технології в освіті та науці*. Мелітополь: МДПУ імені Б. Хмельницького, 2015. Вип. 7. С. 121-127.
5. Олексюк В. Досвід організації віртуальних лабораторій на основі технологій хмарних обчислень. *Інформаційні технології в освіті*. 2014. № 20. С. 128-138.
6. Основы Proxmox VE. URL: <http://onreader.mdl.ru/LearningProxmoxVE/content/Ch01.html> (дата звернення: 04.04.2019).
7. Яшанов С. М. Віртуальні машини в системі інформаційно-навчального середовища вищого закладу освіти. *Інформаційні технології і засоби навчання*. 2010. № 2(16). URL: <http://www.ime.edu-ua.net/em16/emg.htm> (дата звернення: 04.04.2019).

References:

1. Bykov V.YU. Khmarna kompiuterno-tehnolohichna platforma vidkrytoyi osvity ta vidpovidnyy rozvytok orhanizatsiyno-tehnolohichnoyi budovy IT pidrozdiliv navchal'nykh zakladiv. *Teoriya i praktyka upravlinnya sotsialnyimi systemamy: filosofii, psykholohiia, pedahohika, sotsiolohiia*. 2013. № 1. S. 81-98.
2. Virtualizatsiya. URL: <https://uk.wikipedia.org/wiki/Virtualizatsiya>. (data zvernennya: 04.04.2019).
3. Zhaldak M.I. Problemy informatyzatsiyi navchal'noho protsesu v serednikh i vyshchyykh navchal'nykh zakladakh. *Kompiuter u shkoli ta simi*. 2013. № 3. S. 8-15.
4. Naumuk O.V. Stan ta perspektyvy vprovadzhennya zasobiv virtualizatsiyi u protses vyvchennya dystsypliny «Administruvannya komp'yuternykh merezh» *Informatsiyi tekhnohohiyi v osviti ta nauksi*. Melitopol': MDPU imeni B. Khmel'nyts'koho, 2015. Vyp. 7. S. 121-127.
5. Oleksyuk V. Dosvid orhanizatsiyi virtual'nykh laboratoriy na osnovi tekhnohohiy khmarnykh obchyslen'. *Informatsiyi tekhnohohiyi v osviti*. 2014. № 20. S. 128-138.
6. Osnovy Proxmox VE. URL: <http://onreader.mdl.ru/LearningProxmoxVE/content/Ch01.html> (data zvernennya: 04.04.2019).
7. Yashanov S. M. Virtual'ni mashyny v systemi informatsiyno-navchal'noho seredovyscha vyshchoho zakladu osvity. *Informatsiyi tekhnohohiyi i zasoby navchannya*. 2010. № 2(16). URL: <http://www.ime.edu.ua/net/em16/emg.htm> (data zvernennya: 04.04.2019).

Using the Proxmox Web-Based Virtual Environment in pedagogical educational institutions

V.M. Franchuk

Abstract. The article discusses tools for creating virtual machines using the Web-based Proxmox VE system using three open virtualization technologies, namely KVM, QEMU, LXC. The use of Proxmox VE in educational institutions can have the benefits of improving the level of service of computer equipment, simplifying data management and protection, reducing the cost of hardware and power supplies, and the benefits of virtualization during development in testing environments. It is the use of the web-based system interface that allows virtual machines to be managed using modern virtualization approaches, therefore, the article describes in more detail the use and creation of virtual machines using open virtualization technologies.

Keywords: virtualization, virtual machinery, containerization.

УДК: 378.147+004.9

Л.В. Брескіна

кандидат педагогічних наук, доцент;

О.І. Шувалова

старший викладач

Південноукраїнський національний педагогічний університет імені К.Д. Ушинського

Фундаменталізація змістового компоненту навчання веб-програмування майбутніх учителів інформатики

Анотація. Стаття присвячена обґрунтуванню підходу до формування змістового компонента методичної системи навчання Web-програмування. В роботі наводяться критерії фундаменталізації навчання майбутніх учителів інформатики та досліджується питання як зробити навчання таким, що відповідає цим критеріям. На основі визначених загальних підходів в роботі деталізуються особливості фундаменталізації навчання Web-програмування (визначаються інформатичні компетентності, що формуються в ході опанування Web-програмування під час реалізації фундаментального підходу до навчання). Надається опис змістового компоненту методичної системи навчання Web-програмування, що формує виокремлені інформатичні компетентності. Ефективність розробленої методичної системи перевірена педагогічним експериментом, що описаний у роботі.

Ключові слова: Web-програмування, підготовка майбутніх учителів інформатики, фундаменталізація навчання.

Сучасна підготовка фахівців у навчальних закладах потребує особливої уваги до одержання досвіду відповідної діяльності, як до основного компонента формування системи компетентностей в певній галузі [4]. Говорячи про підготовку майбутніх учителів інформатики, виникає проблема відповідності комплексу знань, умінь, навичок та досвіду певної діяльності сучасному рівню розвитку інформатики та інформаційних технологій. Тут необхідно звернутися до дуальної системи у підготовці майбутніх учителів інформатики, що передбачає узгоджену взаємодію освітньої та виробничої сфери [1]. Виробничою сферою для викладача інформатики є і робота у школі (з усіма її організаційними, методичними та виховними аспектами), і робота розробників програмного забезпечення (з її технічними та технологічними аспектами). Саме другий напрям зв'язку з виробничою