Операційні системи

Навчально-методичний посібник

Дорош А.Б.

Чернівці
Чернівецький національний університет
імені Юрія Федьковича
2025

УДК 004.451(075.8)
О-609

Друкується за ухвалою Вченої Ради факультету математики та інформатики Чернівецького національного університету імені Юрія Федьковича, протокол №10 від 23.04.2025 р.

Рецензенти:
І.Д.Скутар, кандидат фіз.-мат. наук, асистент кафедри прикладної математики та інформаційних технологій Факультету математики та інформатики Чернівецького національного університету імені Юрія Федьковича
В.В.Лазорик, кандидат фіз.-мат. наук, доцент кафедри математичних проблем управління і кібернетики Навчально-наукового інституту фізико-технічних та комп'ютерних наук Чернівецького національного університету імені Юрія Федьковича

О-609 Операційні системи. Навчально-методичний посібник / Укл.: Дорош А.Б. Чернівці: Чернівецький нац. ун-т, 2025. 110 с.

Посібник містить теоретичний матеріал, приклади розв'язування типових завдань, завдання для лабораторних занять із курсу операційних систем. Для здобувачів вищої освіти першого (бакалаврського) рівня спеціальностей 122 «Комп'ютерні науки», 124 «Системний аналіз» та 014.09 «Середня освіта (Інформатика)».

Зміст

Розділ 1

Виникнення та розвиток операційних систем

1.1 Від найдавніших комп'ютерів до 1991 року

Спочатку були комп'ютери, які взагалі обходились без ОС. На них запускалась одна програма яка завантажувалася з перфокарти.

Із ростом швидкодії комп'ютерів час роботи програм став на порядок меншим. Черга на виконання перетворилася з буквальної черги програмістів перед дверима операційного залу в ряди котушок з магнітними стрічками, та стовпчики перфокарт.

Попередниками операційних систем слід вважати службові програми (завантажувачі й монітори), а також бібліотеки часто використовуваних підпрограм.

Службові програми мінімізували фізичні маніпуляції оператора з обладнанням, а бібліотеки дозволяли уникнути багаторазового програмування одних і тих же дій

(здійснення операцій введення-виведення, обчислення математичних функцій і т.п.).

Ці бібліотеки стали невидимим ПЗ, яке запускалось перед першою задачею користувача, і керувало її завантаженням та виконанням, виділяло ресурси, записувало результати роботи, слідкувало за правильним завершенням та звільненням ресурсів і негайно після цього переходило до наступної задачі.

Такі фонові програми, ще до впровадження терміну ОС, називались «моніторами».

Упродовж 1969-1970-х років група співробітників підрозділу Bell Labs корпорації AT&T у складі Кена Томпсона, Денніса Рітчі та Дугласа Макілроя розробила операційну систему Unix.

Наразі існує безліч різновидів UNIX-систем, які, в свою чергу, об'єднуються в родини. У їх розробці в різний час брали участь AT&T, деякі комерційні фірми, а також некомерційні організації.

Девіз Unix: «Живи вільним або помри» (англ. «Live Free or Die»).

У 1969 році компанія Honeywell випускає «Кухонний Комп'ютер» H316 — перший домашній комп'ютер (вартістю $10 600).

У 1973 році було випущено Xerox Alto, перший персональний комп'ютер із графічним інтерфейсом.

У 1974 році компанія MITS почала виробництво комп'ютера Altair 8800, який, як вважається, дав початок усім любительським персональним комп'ютерам.

У 1975 році компанія MOS Technology, Inc. почала виробництво комп'ютера KIM-1, який за вартості $245 був простішим порівняно з популярним та більш дорогим Altair 8800, що зробило його дуже популярним серед радіолюбителів та ентузіастів.

У 1976 році Стів Джобс та Стів Возняк почали паяти в гаражі у Стіва Джобса комп'ютери Apple I.

У 2015 році на Інтернет-аукціоні Ebay було продано один із перших комп'ютерів Apple (у робочому стані), зібраний у гаражі Стіва Джобса.

У 1977 році з'явився Commodore PET — перший комп'ютер, до комплекту якого входили клавіатура, монітор і накопичувач на магнітній стрічці (спеціальний фірмовий магнітофон).

У 1979 році почався масовий продаж домашніх персональних комп'ютерів Atari 400/800, які по суті були продовженням популярної ігрової приставки Atari 2600.

У 1980 році у TRS-80 Color Computer вперше у ПК використано багатокористувацьку та багатозадачну операційну систему OS-9.

Також у 1980 році було випущено Sinclair ZX80 — перший персональний комп'ютер для домашнього використання вартістю менше 100 англійських фунтів.

А далі йшли революційні роки в комп'ютерних технологіях...

У 1981 році було випущено Xerox 8010 Star, перший комп'ютер із мишею. Цей комп'ютер використовував операційну систему Pilot, створену Xerox PARC (підрозділ Xerox).

У 1981 компанією Microsoft випущено операційну систему MS-DOS.

12 серпня 1981 року компанія IBM представила широкій публіці першу модель персонального комп'ютера IBM PC 5150, що став фактично родоначальником сучасних комп'ютерів з архітектурою Intel x86.

IBM PC 5150 використовував операційну систему PC DOS — різновид MS-DOS, створений для IBM.

У квітні 1982 році вийшов ZX Spectrum — найпопуляр-

ніший британський комп'ютер, який розвинув комп'ютерну індустрію в Сполученому Королівстві.

Вартість ZX Spectrum була лише 125 фунтів за модель із 16 КБ пам'яті, та 175 фунтів за модель із 48 КБ. Низькій вартості сприяли використання побутового касетного магнітофона як зовнішнього накопичувача, та телевізійного приймача як монітора.

За заслуги в розвитку суспільства (не лише виробництво комп'ютерів) засновник компанії Sinclair Research сер Клайв Сінклер був нагороджений дворянським званням «Лицар королівського Ордену».

У 1986 році вийшов Zx Spectrum +2, котрий мав 128 КБ пам'яті та вбудований касетний магнітофон. Крім цього, він мав 2 порти для підключення джойстика.

ZX Spectrum +3 у 1987 році став наступною моделлю комп'ютера, але містив уже вбудований дисковод для дисків розміром 3 дюйми (не плутати із 3,5 дюйма).

Популярність ZX Spectrum у Європі призвела до появи численних клонів, що створювались як любителями, так і невеликими фірмами.

У Радянському Союзі ZX Spectrum з'явився у кінці 1980-х рр., і майже відразу з'явились клони, випущені спочатку на оригінальному процесорі, а потім і на радянських аналогах Т34ВМ1 та КР1582ВМ1. У 1984-1985 роках був створений також львівський клон ZX Spectrum. В інших країнах створювалися власні клони, наприклад, у Бразилії.

Більшість сучасних пристроїв мають достатню потужність для емуляції процесора Z80. Завдяки великій базі готового програмного забезпечення (переважно ігор) «Sinclair ZX Spectrum» є, можливо, найпопулярнішою платформою, що емулюється у світі. Існує понад півтори сотні емуляторів для всіх поширених операційних систем

від Windows та Linux до Android та iOS.

31 серпня 1999 року компанія Amstrad дозволила поширювати вміст постійної пам'яті для комп'ютерів ZX Spectrum, тому емуляція цього сімейства є легальною.

У серпні 1982 року з'явився Commodore 64 — найбільш продаваний американський комп'ютер того часу: продано понад 20 мільйонів штук.

Стартовий екран Commodore 64 було використано у заставці гри 2002 року GTA: Vice City.

У 1982 році компаніями Sony та Philips розпочато випуск перших компакт-дисків (CD). Спершу компакт-диски створювались для зберігання аудіо та музики, проте згодом його почали використовувати для зберігання будь-яких даних у двійковій формі.

До речі, перший в історії музичний альбом, виданий на CD, був «The Visitors» гурту ABBA.

У березні 1983 року компанія Compaq розпочала продаж Compaq Portable — першого портативного комп'ютера, а також першого клону комп'ютерів серії IBM PC.

У січні 1984 року надійшов у масове виробництво перший персональний комп'ютер із мишею та повністю графічним інтерфейсом. Його назва — Apple Macintosh.

У 1985 році компанія Microsoft випускає нову операційну систему — Windows.

У 1986 році з'явився перший ноутбук IBM PC Convertible від компанії IBM.

У 1991 році фінський студент Лінус Торвальдс вирішив поекспериментувати з командами процесора Intel 80386 і те, що вийшло, виклав у Інтернет. Сотні програмістів із різних країн світу стали дописувати й переробляти програму. Вона перетворилась на повнофункціональну операційну систему.

За нею закріпилась назва Linux — внаслідок поєдна-

ння двох слів: «Linu» від імені розробника, а «x» — від назви іншої ОС UNIX, оскільки нова ОС була дуже на неї схожою, от тільки тепер працювала і на комп'ютерах з архітектурою x86.

При переході в еру персональних комп'ютерів відбувся зсув в розумінні поняття операційної системи.

Першим автомобілям бракувало спідометрів, радіо та склоочисників, які згодом стали стандартними компонентами, так само першим ОС бракувало багато необов'язкових функцій (наприклад текстові редактори, файлові менеджери і т.д.), які на ПК стали обов'язковими компонентами ОС.

Сучасний користувач навіть не уявляє ОС без графічного інтерфейсу. А справжнім нащадком перших ОС є те, що сьогодні називається ядром.

1.2 Unix та Linux

У середині 50-х років дослідницький відділ корпорації Bell System придбав для свого комп'ютерного центра кілька нових комп'ютерів.

У середині 50-х років дослідницький відділ корпорації Bell System придбав для свого комп'ютерного центра кілька нових комп'ютерів.

Величезні машини, куплені за мільйони доларів в IBM, призначалися для проведення розробок у поки ще мало вивченій комп'ютерній області.

Але коли вчені Bell освоїлися на встановлених мейнфреймах, стало ясно, що програмне забезпечення, яке йде в поставці, зовсім не підходить для повноцінної дослідницької роботи.

А через відсутність операційної системи все доводилося робити вручну, що забирало багато часу й сил.

Оскільки більшість співробітників відділу були досвідченими програмістами, вони вирішили створити ОС самотужки. І втілити в ній усе, що їм було потрібно.

Спільними зусиллями розробка системи BESYS (Bell Operating System) зайняла менше року, і в 1957 р. вона була встановлена на всіх машинах комп'ютерного центра.

У 1964 р. Bell Labs придбала нове дороге обладнання, включаючи потужніші комп'ютери, встановлені в дослідницькому відділі.

Програмісти компанії знову зіштовхнулися зі старою проблемою.

BESYS була заточена під конкретну платформу й не працювала на нових машинах, а IBM, як і раніше, мало піклувалася про програми, займаючись створенням винятково заліза. Залишалося розраховувати тільки на себе.

Було прийнято рішення не розробляти нову власну операційну систему, а підключитися до сумісного проекту General Electric та Массачусетського технологічного інституту MULTICS.

Вагому підтримку проекту надав телекомунікаційний велетень AT&T, але у 1969 році він вийшов з проекту, оскільки MULTICS не приносила жодних фінансових вигод.

Упродовж 1969-1970-х років група співробітників Bell Labs у складі Кена Томпсона, Денніса Рітчі та Дугласа Макілроя розробила операційну систему Unix.

Спочатку система мала назву UNICS (UNIplexed Information and Computing System). Пізніше ця назва скоротилась до UNIX.

Девіз Unix: «Живи вільним або помри» (англ. «Live Free or Die»).

У 1973 р. було прийнято рішення переписати ядро системи на щойно створеній мові C.

UNIX стала першою операційною системою, майже повністю написаною мовою програмування високого рівня, що суттєво спростило портування системи на інші архітектури.

До 1978 р. система використовувалась більш ніж на 600 машинах, перш за все, в університетах.

Версія 7 було останньою єдиною версією UNIX.

На початку 1980-х років компанія AT&T, якій належала Bell Labs, зрозуміла цінність UNIX і почала створення комерційної версії UNIX.

Ця версія надійшла до продажу в 1982 році.

Трохи раніше, у 1977 р. лабораторія Білла Джоя в університеті Берклі створила власну версію UNIX, яка базувалась на UNIX Version 6. Ця версія отримала назву BSD (англ. Berkeley Software Distribution).

Незабаром розробка BSD була продовжена у декількох напрямах одночасно, що призвело до появи проектів, відомих під назвами FreeBSD, NetBSD, OpenBSD та DragonFlyBSD.

Коли компанія Apple шукала основу для своєї операційної системи, вона вибрала NEXTSTEP — операційну систему з вільно розповсюджуваним ядром, розроблену фірмою NeXT та перейменованою у Darwin після придбання фірмою Apple.

Ця система відноситься до сімейства BSD. Застосування Darwin BSD UNIX у Mac OS X робить її однією з найбільш розповсюджених версій UNIX.

У 1983 році Річард Столлмен оголосив про створення проекту GNU — спроба створення вільної UNIX-подібної операційної системи з нуля, без використання оригінального початкового коду. GNU — рекурсивне скорочення від англ. «GNU's Not Unix».

Більша частина програмного забезпечення, розроблe-

ного в рамках цього проекту — такого, як GNU toolchain, Glibc (стандартна бібліотека мови Сі) та Coreutils — відіграють ключову роль у інших вільних операційних системах.

Однак, роботи зі створення заміни для ядра UNIX, необхідного для повного виконання задач GNU, відбуваються дуже повільно, навіть на сьогоднішній день.

У 1991 році фінський студент Лінус Торвальдс вирішив поекспериментувати з командами процесора Intel 80386 і те, що вийшло, виклав у Інтернет. Сотні програмістів із різних країн світу стали дописувати й переробляти програму. Вона перетворилась на повнофункціональну операційну систему. За нею закріпилась назва Linux — внаслідок поєднання двох слів: «Linu» від імені розробника, а «x» — від назви UNIX, оскільки нова ОС була дуже на неї схожою.

Логотипом Linux є пінгвін Тукс.

Першу версію ядра Linux було опубліковано в Інтернеті 17 вересня 1991 року.

Об'єднавшись із ядром Linux, програмне забезпечення GNU стало основою для UNIX-подібної операційної системи, відомою як Linux.

Дистрибутиви цієї системи, які включають ядро, службові програми GNU та додаткове програмне забезпечення стали популярними як серед аматорів, так і серед фахівців.

Дистрибутив (англ. distribute — розповсюджувати) — форма розповсюдження програмного забезпечення.

Дистрибутив зазвичай містить програму-встановлювач (для вибору режимів і параметрів встановлення), програми для початкової ініціалізації системи, службові файли та прикладні програми.

У випадку Linux, робота над розробкою ядра ведеться

однією групою програмістів, базових користувацьких програм іншою (наприклад, проектом GNU), і усе це збираються третіми групами у дистрибутив.

Одним із найдавніших дистрибутивів Linux є Debian. Перша версія вийшла у 1993 р., а найновіша — 11.5 (Bullseye) у 2022 р.

Із Debian утворилося декілька похідних дистрибутивів, наприклад, Knoppix, Ubuntu.

Іншу важливу гілку утворює комерційний дистрибутив Red Hat Enterprise Linux (RHEL). Перша версія вийшла у 2000 р.

Є також безкоштовні дистрибутиви на базі RHEL — наприклад, CentOS, Fedora.

Виникає питання: стільки дистрибутивів ОС і у всіх треба працювати в консольному режимі?

Аж ніяк. На щастя, знайшлися програмісти, які подумали про це і створили графічні оболонки, тобто візуальні середовища (desktop environments) цих систем.

1.3 Графічний інтерфейс користувача

Інтерфейс користувача (ІК), (англ. user interface, UI) — засіб зручної взаємодії користувача з інформаційною системою.

Основні різновиди інтерфейсу користувача:

Текстовий інтерфейс користувача (англ. Text user interface, TUI) — спосіб взаємодії користувача з комп'ютером з використанням текстового (буквено-цифрового) режиму дисплея або аналогічних — наприклад, командного рядка.

Графічний інтерфейс користувача (ГІК, англ. Graphical user interface, GUI) — тип інтерфейсу, який дозволяє користувачам взаємодіяти з електронними пристроями через графічні зображення та візуальні вказівки.

Інтерфейс командного рядка (англ. command-line interface, CLI) — різновид текстового інтерфейсу користувача й комп'ютера, в якому інструкції комп'ютеру можна дати тільки введенням із клавіатури текстових рядків (команд). Також відомий під назвою консоль.

Вперше концепція GUI була запропонована вченими з дослідницької лабораторії Xerox PARC у комп'ютері Xerox Alto в 1970-х рр.

Комерційне втілення GUI отримав лише в продуктах корпорації Apple Computer, зокрема Lisa та Macintosh.

Зараз GUI є стандартом для більшості доступних на ринку операційних систем і додатків.

Окрім комп'ютерів, GUI використовується в мобільних пристроях, таких, як мобільні телефони, планшети, електронні книги, портативні медіаплеєри тощо.

Найпоширеніші візуальні оболонки для UNIX-подібних систем:

- GNOME (GNU Network Object Model Environment). Використовується в Ubuntu з 18.04.
- Unity — використовувався в Ubuntu до версії 18.04.
- KDE — використовується в Kubuntu.
- XFCE — використовується в Xubuntu.
- LXDE — використовується в Lubuntu.
- MATE — використовується в Ubuntu MATE. Є продовженням розвитку GNOME2.

- Cinnamon — використовується в Linux Mint.

Є ще чимало інших візуальних оболонок. Здебільшого всі вони мають робочий стіл (стільницю) та вікна. Тому їх називають стільничними середовищами (desktop environments).

Одним із різновидів GUI є масштабований інтерфейс користувача (англ. Zooming User Interface, скор. ZUI). Робочий простір являє собою велику або необмежену площину, на якій розташовані основні елементи, властивості і вміст яких стають доступними в міру їх «наближення» шляхом збільшення. Подальше наближення вмісту робить доступним більш глибокі рівні.

Переваги GUI:

- Графічний інтерфейс є «дружнім» (простим та інтуїтивним) для користувачів, які щойно розпочали знайомство з комп'ютером.
- У програмах для обробки графіки він найчастіше є єдино можливим.

Недоліки:

- Більше споживання пам'яті в порівнянні з текстовим інтерфейсом.
- Складніше організувати дистанційну роботу.
- Неможливість автоматизації, якщо вона не була закладена автором програми.

Розділ 2

Його величність Linux

2.1 Основні дистрибутиви Linux

Наразі під терміном Linux розуміють лише ядро системи, а разом із додатковими програмами та візуальною оболонкою вони утворюють дистрибутиви.

Одним із найдавніших дистрибутивів Linux є Debian. Перша версія вийшла у 1993 р., а найновіша – 12 (Bookworm) у 2023 р.

Офіційний сайт: https://www.debian.org

Наразі Debian використовує графічний інтерфейс GNOME і менеджер пакетів dpkg.

Із Debian утворилося декілька похідних дистрибутивів, наприклад, Knoppix, Ubuntu.

Knoppix — це дистрибутив GNU/Linux зі збіркою вільних програм, які працюють із компакт-диска (Live CD чи Live-DVD). Встановлення його на жорсткий диск не обов'язкове.

Knoppix можна використовувати як демонстраційний CD/DVD у комп'ютерних салонах, для навчання, для відновлення встановленої системи тощо.

Перша версія Knopріх вийшла у 2000 р., а найновіша — 9.3 у 2022 р.

Knopріх використовує графічне середовище LXDE (раніше був KDE).

Менеджер пакетів — dpkg.

Офіційний сайт: http://knopper.net

Ubuntu — найпопулярніший на сьогоднішній день дистрибутив на основі Debian.

Розробник — Canonical Ltd. на чолі з Марком Шатлвортом.

Назва Ubuntu походить із південноафриканської філософії убунту (дослівно перекладається як людяність). Canonical вважає, що це можна трактувати як "людяність для інших".

Перша версія Ubuntu 4.10 вийшла у жовтні 2004 р., а найновіша — 25.04 (Plucky Puffin) у квітні 2025 р.

Нова версія Ubuntu з'являється кожного квітня та жовтня. У квітні кожного парного року виходить довгостроковий реліз — Long Term Support (LTS), який підтримується 5 років. Усі інші проміжні релізи, зазвичай, підтримуються 9 місяців.

Окрім числової, кожна версія Ubuntu має ще й текстову назву, яка складається з 2 слів. Вони починаються з однакових літер.

Наприклад:

4.10 Warty Warthog (Пильний бородавочник)

5.04 Hoary Hedgehog (Сивий їжак)

5.10 Breezy Badger (Бадьорий борсук)

16.04 LTS Xenial Xerus (Гостинний ксерус, рід ссавців з родини вивіркові)

16.10 Yakkety Yak (Балакучий як)

17.04 Zesty Zapus (Пікантний запус, рід північноамериканських ссавців-гризунів)

17.10 Artful Aardvark (Спритний трубкозуб)

18.04 LTS Bionic Beaver (Біонічний бобер)

18.10 Cosmic Cuttlefish (Космічна каракатиця)

19.04 Disco Dingo (Диско дінго)

19.10 Eoan Ermine (Світанковий горностай)

20.04 Focal Fossa (Фокальна фоса, котовидий ссавець)

20.10 Groovy Gorilla (Захоплива горила)

21.04 Hirsute Hippo (Кудлатий гіпопотам)

21.10 Impish Indri (Пустотливий індрі, різновид лемурів)

22.04 Jammy Jellyfish (Джемова медуза)

22.10 Kinetic Kudu (Кінетичний куду, рід антилоп)

25.04 Plucky Puffin (Відважний пухнастик)

Ubuntu 18.04 і новіші використовують графічний інтерфейс GNOME (GNU Network Object Model Environment).

До версії 18.04 використовувався графічний інтерфейс Unity — власна розробка Canonical.

Починаючи з версії 18.04, Ubuntu більше не підтримує 32-розрядні системи, а лише 64-розрядні.

Також Ubuntu має ще серверний різновид Ubuntu Server, без графічного інтерфейсу.

Усі версії Ubuntu використовують той же менеджер пакетів, що й Debian — dpkg.

Офіційний сайт: https://ubuntu.com

Kubuntu — різновид Ubuntu, що використовує графічний інтерфейс KDE.

Розробник — Canonical Ltd.

KDE — використовується в Kubuntu.

Нумерація та назви версій — такі ж як і в Ubuntu.

На відміну від Ubuntu, Kubuntu 18.04 підтримує 32-розрядні системи, а 19.04 вже ні.

Офіційний сайт: https://kubuntu.org

Xubuntu — різновид Ubuntu, що використовує графічний інтерфейс XFCE.

Розробник — Canonical Ltd.

XFCE — використовується в Xubuntu.

Нумерація та назви версій — такі ж як і в Ubuntu.

На відміну від Ubuntu, Xubuntu 18.04 досі підтримує 32-розрядні системи, а 19.04 вже ні.

Офіційний сайт: https://xubuntu.org

Lubuntu — різновид Ubuntu, що використовує графічний інтерфейс LXDE.

Розробник — спільнота Lubuntu та LXDE Foundation.

LXDE — використовується в Lubuntu.

Нумерація та назви версій — такі ж як і в Ubuntu.

Lubuntu 18.04 також підтримує 32-розрядні системи, а 19.04 вже ні.

Офіційний сайт: https://lubuntu.net

Ubuntu MATE — різновид Ubuntu, що використовує графічний інтерфейс MATE.

Розробник — команда Ubuntu MATE.

MATE — використовується в Ubuntu MATE.

Реліз GNOME 3, який мав на меті замінити "класичний"графічний інтерфейс на концептуально новий, викликав шквал критики в середовищі Linux-спільноти. Багато користувачів відмовились ставити новий, істотно видозмінений GNOME, закликаючи водночас продовжити розробку середовища GNOME 2. Проект MATE був ініційований користувачами Arch Linux із метою подальшого розвитку даної благородної місії.

Нумерація та назви версій — такі ж як і в Ubuntu.

Ubuntu MATE 18.04 також підтримує 32-розрядні системи, а 19.04 вже ні.

Офіційний сайт: https://ubuntu-mate.org

Ubuntu Kylin розроблюється спільно компанією Canonical та урядовими організаціями КНР для китайського ринку, враховуючи специфіку китайської мови та популярних в китайському Інтернеті онлайн-сервісів.

Ubuntu Kylin використовує графічний інтерфейс UKUI (на базі MATE і GNOME 2).

Нумерація та назви версій — такі ж як і в Ubuntu.

Ubuntu Kylin 18.04 також підтримує 32-розрядні системи, а 19.04 вже ні.

Офіційний сайт: http://ubuntukylin.com

Linux Mint — серія дистрибутивів на основі Ubuntu.

Розробник — Mint team.

Існує три різновиди Linux Mint, залежно від графічного середовища, яке вони використовують: Linux Mint Cinnamon, Linux Mint MATE, Linux Mint Xfce.

Нумерація та назви версій не такі, як в Ubuntu. Вони базуються на жіночих іменах.

1.0 Ada
2.0 Barbara
2.1 Bea
2.2 Bianca
3.0 Cassandra
17.1 Rebecca
17.2 Rafaela
17.3 Rosa
18.0 Sarah
18.1 Serena
18.2 Sonya
19.0 Tara
19.1 Tessa
19.2 Tina
19.3 Tricia
20 Ulyana

20.1 Ulyssa
20.2 Uma
20.3 Una
21 Vanessa
21.1 Vera
21.2 Victoria
21.3 Virginia
22 Wilma
22.1 Xia

Linux Mint досі підтримує 32-розрядні системи.

Використовується менеджер пакетів dpkg.

Офіційний сайт: https://linuxmint.com

Свого часу була навіть спроба створити український дистрибутив Linux. Його назва – Grusha Linux. Найновіша версія 3.4 була випущена у 2011 р. Графічні інтерфейси: KDE, GNOME, LXDE. Менеджер пакетів — dpkg.

Kali Linux — це дистрибутив Debian-похідних Linux, призначений для цифрової криміналістики і тестування на проникнення.

Перша версія вийшла у 2013 р., а найновіша – 2025.1a у 2025 р.

Розробник: Offensive Security.

Kali Linux використовує графічний інтерфейс GNOME.

Kali Linux має понад 300 встановлених програм для тестування на проникнення.

Kali Linux може працювати при установці на жорсткий диск комп'ютера або завантажившись із Live CD чи USB-носія. Також він може працювати у віртуальній машині.

Kali Linux використовується багатьма хакерами у фільмах та серіалах.

Kali Linux підтримує 32- і 64-розрядні платформи.

Менеджер пакетів — dpkg.

Офіційний сайт: https://kali.org

Іншу важливу гілку утворює комерційний дистрибутив Red Hat Enterprise Linux (RHEL). Перша версія вийшла у 2000 р., а найновіша — 10.0 у 2024 р.

RHEL використовує графічний інтерфейс GNOME.

Розробник — Red Hat.

Менеджер пакетів — yum, а з версії 8 — dnf.

Офіційний сайт: https://redhat.com

Fedora — безкоштовний дистрибутив на базі RHEL.

Перша версія вийшла у 2003 р., а найновіша – 42 у 2025 р.

Розробник: Fedora Project. Спонсор: Red Hat.

Fedora використовує графічний інтерфейс GNOME 3.

Fedora є відгалуженням від дистрибутиву Red Hat Linux, і має намір стати його заміною для домашніх та офісних комп'ютерів.

Проєкт служить для тестування нових технологій, які надалі включаються в продукти Red Hat і інших виробників.

Проект Fedora отримав ім'я за логотипом компанії Red Hat, на якому присутній червоний капелюх-федора.

Fedora, зокрема, використовує на більшості своїх машин автор ядра Linux – Лінус Торвальдс.

Fedora має також і версію для серверних комп'ютерів.

Менеджер пакетів — dnf.

Офіційний сайт: https://getfedora.org

CentOS (Community ENTerprise Operating System) — безкоштовний дистрибутив на базі комерційного Red Hat Enterprise Linux.

Перша версія вийшла у 2004 р., а найновіша — 8.5 у 2021 р.

Розробник: CentOS.

Red Hat Enterprise Linux складений переважно з вільного та відкритого програмного забезпечення, але наявний у доступній для використання, двійковій формі лише для передплатних користувачів.

Як і вимагається, Red Hat випускає усі сирцеві тексти своїх продуктів під GNU General Public License та іншими вільними ліцензіями.

Розробники CentOS використовують цей сирцевий код для створення кінцевого продукту, котрий є дуже подібним до Red Hat Enterprise Linux і вільним для завантаження та використання, однак без відповідної технічної підтримки з боку компанії Red Hat.

Існують й інші дистрибутиви, що базуються на сирцевих текстах Red Hat Enterprise Linux, однак жоден з них не досяг такого рівня спільноти.

Загалом CentOS — єдиний дистрибутив, котрий йде у ногу із змінами, що вносяться до Red Hat Enterprise Linux.

CentOS використовує графічний інтерфейс GNOME.

Починаючи з версії 7, CentOS підтримує лише 64-розрядні системи.

Менеджер пакетів — yum.

Офіційний сайт: https://centos.org

Arch Linux — мінімалістичний, гнучкий дистрибутив Linux.

Перша версія вийшла у 2002 р., а найновіша — у 2022 р.

Розробник: Аарон Ґріффін і команда.

Як гадаєте, який графічний інтерфейс постачається в складі Arch Linux?

Довгий час за замовчуванням був консольний режим. Проте на Arch Linux можна встановити більшість відомих візуальних оболонок.

Офіційно Arch Linux підтримує лише 64-розрядні системи.

Arch Linux має власний менеджер пакетів — pacman.

Офіційний сайт: https://archlinux.org

2.2 Основні консольні команди Linux

Консоль (англ. console) — пристрій, який забезпечує взаємодію оператора комп'ютера з операційною системою.

Як правило, як консоль використовується дисплей і клавіатура, або окремий комп'ютерний термінал.

Також під словом консоль часто мають на увазі інтерфейс командного рядка.

Інтерфейс командного рядка (англ. command-line interface, CLI) — різновид текстового інтерфейсу користувача й комп'ютера, в якому інструкції комп'ютеру можна дати тільки введенням із клавіатури текстових рядків (команд).

whoami – виводить на екран логін, під яким працює поточний користувач

pwd – показує шлях до директорії, в якій зараз знаходиться поточний користувач

ls – показує список файлів та директорій у поточній директорії (окрім прихованих файлів)

ls -al – те саме, що ls, тільки показуються приховані файли, а також власник кожного файлу і права доступу

cd шлях_до_директорії – переходить до вказаної директорії

Команда cd розуміє такі скорочення:

. (крапка) — поточна директорія

.. (дві крапки) — батьківська директорія

~(тильда) — домашня директорія поточного користувача

mkdir шлях_до_директорії – створює нову директорію

touch шлях_до_файлу – створює порожній файл

nano шлях_до_файлу – відкриває файл для редагування у консольному текстовому редакторі nano. Якщо файлу не існує, його буде створено.

cp звідки куди – копіює файл/директорію

mv звідки куди – переміщує файл/директорію

rm шлях_до_файлу – видаляє існуючий файл

rm -rf шлях_до_директорії – видаляє існуючу директорію

cat шлях_до_файлу – показує вміст файлу на екран

Наприклад:

cat /etc/passwd – виведе вміст файлу /etc/passwd (список усіх користувачів)

Пошук у файлі:

cat /etc/passwd | grep root – виведе лише ті рядки файлу, в яких зустрічається слово root

sudo chown користувач:група шлях – змінює власника існуючого файлу чи директорії

sudo chown -R користувач:група шлях_до_директорії – змінює власника директорії та всіх файлів у ній

Наприклад:

sudo chown admin:admin a.txt – власником файлу a.txt стане користувач admin та група admin

sudo chmod права шлях_до_файлу_чи_директорії – змінює права доступу до існуючого файлу чи директорії

sudo chmod -R права шлях_до_директорії – змінює права доступу до директорії та всіх файлів у ній

Права доступу:

4 — читання

2 — редагування

1 — виконання

Наприклад: chmod 644 a.txt – власник файлу може читати і редагувати його; група власника може лише читати; усі інші користувачі можуть лише читати

Наприклад: chmod 755 /tmp – власник директорії може переглядати її, редагувати файли в ній та запускати бінарні файли на виконання; група власника може лише переглядати файли в директорії та запускати бінарні файли в ній; усі інші користувачі можуть лише переглядати файли в директорії та запускати бінарні файли в ній

Команда ls -al друкує права доступу і власника кожного файла й папки. Але права доступу показуються літерами, а не цифрами.

r = 4 (reading, читання)

w = 2 (writing, редагування)

x = 1 (execution, виконання)

d означає, що це директорія

man назва_команди – показує довідку (manual) стосовно вибраної команди

Встановити новий пакет або оновити встановлений раніше можна однією з команд (залежно від дистрибутиву Linux):

sudo apt-get install пакет

sudo yum install пакет

sudo pacman -S пакет

Просканувати доступні репозиторії:

sudo apt-get update

sudo yum update

sudo pacman -Syu

Видалити встановлений пакет:

sudo apt-get remove пакет

sudo yum remove пакет

sudo pacman -R пакет

sudo useradd логін – створити нового користувача із вказаним логіном

sudo passwd логін – задати пароль для користувача із вказаним логіном

Наприклад:

sudo useradd user1

sudo passwd user1

Створиться користувач та група user1. Створиться домашній каталог /home/user1. Користувач user1 з'явиться у файлі /etc/passwd.

Інший приклад:

sudo useradd -e 2020-12-31 user1 – акаунт буде дійсним лише до вказаної дати

sudo userdel логін – видалити користувача із вказаним логіном

sudo groupadd назва_групи – створити нову групу користувачів

sudo groupdel назва_групи – видалити групу користувачів

sudo usermod -a -G назва_групи ім'я_користувача – додати користувача до певної групи

groups – показати всі групи, до яких належить поточний користувач

groups ім'я_користувача – показати всі групи, до яких належить вказаний користувач

sudo reboot – перезавантажити комп'ютер

sudo poweroff – вимкнути комп'ютер

Розділ 3

Копаємо глибше

3.1 Процеси в Linux

Операційні системи сімейства **Linux**, як і інші UNIX-подібні системи, побудовані навколо потужної концепції процесів, ниток (threads) та механізмів взаємодії між ними. Завдяки відкритому коду та широкій підтримці спільноти, Linux став універсальною платформою для вивчення основних понять багатозадачності та синхронізації.

Процес — це виконувана програма. У Linux кожен процес має унікальний ідентифікатор (PID), окрему область пам'яті, стек, контекст виконання і власні ресурси. Нові процеси створюються за допомогою системного виклику `fork()`, який копіює поточний процес, або `exec()`, що замінює вміст процесу новою програмою.

У системах на базі ядра Linux *процес* є фундаментальною одиницею виконання. Він репрезентує активний екземпляр програми, що виконується. Процес має власний *адресний простір*, *регістри процесора*, *таблицю відкритих файлів*, *ідентифікатори* та інші ресурси, які керуються ядром операційної системи.

3.1.1 Основні характеристики процесу в Linux

У системах на базі ядра Linux *процес* є фундаментальною одиницею виконання. Він репрезентує активний екземпляр програми, що виконується, і має низку ключових характеристик:

Адресний простір — кожен процес у Linux має власний віртуальний адресний простір. Це означає, що процес працює у своєму ізольованому середовищі пам'яті. Адресний простір поділяється на сегменти:

- `text` — область з виконуваним кодом програми (read-only);
- `data` — змінні з початковими значеннями;
- `bss` — незініціалізовані змінні;
- `heap` — область для динамічно виділеної пам'яті (за допомогою функції `malloc`);
- `stack` — стек викликів, який зростає у протилежному напрямку від heap.

Регістри процесора — при виконанні процесу ядро зберігає поточний стан регістрів процесора у структурі `task_struct`. При перемиканні контексту ядро зберігає регістри активного процесу і відновлює регістри іншого.

Таблиця відкритих файлів — кожен процес має власну таблицю відкритих файлових дескрипторів. Вона є посиланням на структури ядра, які відображають відкриті файли, сокети, канали, тощо. Стандартні дескриптори:

- `0` — `stdin` (вхід),

- `1` — `stdout` (вивід),
- `2` — `stderr` (помилки).

Нові файли відкриваються наступними номерами.

Ідентифікатори — процес має унікальні числові ідентифікатори:

- `PID` (*Process ID*) — унікальний номер процесу;
- `PPID` — PID батьківського процесу;
- `UID`, `EUID` — ідентифікатори користувача;
- `GID`, `EGID` — групові ідентифікатори.

Ці ідентифікатори використовуються ядром для контролю доступу до ресурсів.

Стан процесу — процес може перебувати у різних станах:

- `R` — виконання або готовність до нього;
- `S` — сон (чекає на подію);
- `D` — незворотний сон (наприклад, очікування диска);
- `Z` — *зомбі*-процес (виконання завершене, але батьківський процес не зчитав статус завершення);
- `T` — зупинений (наприклад, за сигналом `SIGSTOP`).

Ці характеристики формують контекст процесу, який зберігається ядром у структурі `task_struct`. Коли відбувається *перемикання контексту*, ядро зберігає стан одного процесу та відновлює інший, дозволяючи таким чином реалізовувати **багатозадачність**.

3.1.2 Створення процесів

У Linux нові процеси створюються за допомогою системного виклику `fork()`. Цей виклик створює новий процес, що є копією батьківського. Новостворений процес називається *дочірнім*, а той, що викликав `fork()`, — *батьківським*.

```
#include <unistd.h>
pid_t pid = fork();
if (pid == 0) {
    // код дочірнього процесу
} else if (pid > 0) {
    // код батьківського процесу
}
```

Дочірній процес отримує копію пам'яті батьківського, але з появою механізму *Copy-on-Write (COW)* фізичне копіювання сторінок відбувається лише при зміні.

3.1.3 Заміна образу процесу

Після створення процесу за допомогою `fork()`, часто викликається один із варіантів функції `exec()` (наприклад, `execl`, `execvp`), щоб завантажити нову програму в поточний процес. У результаті оригінальний код і дані процесу замінюються новими.

```
execl("/bin/ls", "ls", "-l", NULL);
```

3.1.4 Очікування завершення процесу

Батьківський процес може очікувати завершення дочірнього за допомогою виклику `wait()` або `waitpid()`:

```
#include <sys/wait.h>
int status;
wait(&status);
```

3.1.5 Стан процесів

Процес у Linux може перебувати в кількох *станах*:

- `R` — *Running* або готовий до виконання (у черзі на планування);
- `S` — *Sleeping* (очікує події);
- `D` — *Uninterruptible sleep* (часто при I/O);
- `Z` — *Zombie* (завершився, але не зібраний батьком);
- `T` — *Stopped* (зупинений сигналом).

Для перегляду інформації про процеси можна використовувати команду `ps`, наприклад:

```
ps aux | grep myprocess
```

Ієрархія процесів. Процеси організовані в ієрархію: кожен має батьківський (parent) та, можливо, дочірні процеси. Процес `init` (або `systemd` у сучасних системах) є предком усіх інших процесів.

```
pstree
```

Linux надає різні механізми взаємодії процесів:

- `pipes` – канали для передачі даних між родинними процесами;

- `signals` – сигнали для обміну подіями (наприклад, `SIGKILL`, `SIGTERM`);
- `shared memory` – спільна пам'ять (через `shmget`, `mmap`);
- `message queues` і `semaphores` – для синхронізації та передачі повідомлень.

Бувають процеси, що завершились, але ще не були оброблені батьківським. Вони все ще займають запис у таблиці процесів. Коли батьківський процес завершується раніше за дочірній, останній стає *осиротілим* і автоматично передається на обробку процесу `init/systemd`.

3.1.6 Нитки (Threads)

Нитка – це легкий процес, який виконується в рамках одного процесу, використовуючи спільну пам'ять. У Linux нитки створюються за допомогою системного виклику `clone()`.

Переваги ниток:

- спільне використання пам'яті забезпечує швидку взаємодію;
- менші накладні витрати в порівнянні з процесами.

3.1.7 Семафори

Семафори — це механізми синхронізації, які дозволяють уникати конфліктів при одночасному доступі до спільних ресурсів.

- **Бінарний семафор** – два стани: зайнято/вільно.

- **Лічильний семафор** – дозволяє доступ до ресурсу кільком потокам одночасно (до певного ліміту).

Linux підтримує як **System V IPC** семафори, так і **POSIX**-семафори (`sem_init`, `sem_wait`, `sem_post`).

3.1.8 Канали (Pipes)

Канали – механізм обміну даними між процесами:

- `pipe()` створює анонімний канал між батьківським і дочірнім процесом.
- `mkfifo()` створює іменований канал (FIFO), доступний різним процесам.

Один процес записує в канал, інший читає, що забезпечує просту однонаправлену комунікацію.

3.2 Багатозадачність у Linux

Операційна система Linux підтримує *багатозадачність* (*multitasking*), що дозволяє одночасно виконувати кілька процесів. У сучасних системах це досягається завдяки розподілу ресурсів центрального процесора між різними задачами так, щоб користувач мав ілюзію їх паралельного виконання.

3.2.1 Витісняюча багатозадачність

Linux реалізує **витісняючу багатозадачність** (*preemptive multitasking*). Це означає, що ядро самостійно контролює, який процес отримує доступ до процесора, і може

примусово призупинити (витіснити) поточний процес, щоб надати CPU іншому.

Такий підхід дозволяє гарантувати справедливий розподіл часу між процесами та уникати ситуацій, коли один процес блокує виконання інших (наприклад, через помилки або навмисне зловживання).

3.2.2 Планувальник процесів (Scheduler)

Ядро Linux містить *планувальник* (*scheduler*), який відповідає за вибір наступного процесу для виконання. Основними його завданнями є:

- визначити, які процеси готові до виконання (у стані `R` — *Running/Ready*);
- обрати серед них той, що має найвищий *ефективний пріоритет*;
- перемкнути контекст (*context switch*) з поточного процесу на обраний.

Планування виконується на основі:

- *пріоритетів* — кожен процес має значення пріоритету, яке може бути змінено (наприклад, за допомогою команди `nice`);
- *політик планування* (`SCHED_NORMAL`, `SCHED_ FIFO`, `SCHED_RR` тощо);
- *використаного часу процесора* — щоб не допустити монополії ресурсу.

3.2.3 Політики планування

Linux підтримує кілька *політик планування*, які визначають, як саме процеси будуть витісняти один одного:

- `SCHED_OTHER` (або `SCHED_NORMAL`) — звичайна політика для користувацьких процесів із динамічним пріоритетом;
- `SCHED_FIFO` — реального часу, без витіснення (процес виконується, поки сам не звільнить процесор або не буде заблокований);
- `SCHED_RR` — реального часу, з витісненням і круговим плануванням (Round-Robin).

Політику можна перевірити або встановити за допомогою інструменту `chrt`.

```
chrt -p <PID>
chrt -r -p 10 <PID>
% встановлення SCHED_RR з пріоритетом 10
```

3.2.4 Контекстне перемикання (Context switch)

Коли ядро перемикає процеси, воно виконує *контекстне перемикання* — збереження стану поточного процесу (регістри, лічильник команд, тощо) і завантаження стану нового. Це операція з певними накладними витратами, тому планувальник прагне мінімізувати частоту таких перемикань без втрати чуйності системи.

Багатозадачність є ключовою функцією Linux, що дозволяє виконувати багато програм одночасно, ефективно використовуючи апаратні ресурси. *Витісняюча модель* та гнучка система *пріоритетів* і *політик планування* роблять Linux придатним як для інтерактивних систем, так і для систем реального часу.

3.3 Жорсткі та символічні посилання

Linux підтримує два типи файлових посилань:

- **Жорстке посилання** (*hard link*) — це додаткове ім'я для того самого файлу. Усі жорсткі посилання на файл посилаються на один і той самий *inode* — структуру даних, що містить інформацію про файл (права доступу, власника, час створення, місце розташування на диску тощо). Дані зберігаються до тих пір, поки хоча б одне жорстке посилання залишається.

- **Символічне посилання** (*symbolic link, symlink*) — це спеціальний файл, який містить шлях до іншого файлу або директорії. Воно працює подібно до ярлика у Windows. Символічне посилання має свій власний *inode*, який вказує на шлях до цільового файлу.

3.3.1 Створення посилань

Щоб створити жорстке посилання, використовується команда `ln` (від слова *link*):

```
ln original.txt hardlink.txt
```

Щоб створити символічне посилання, використовується параметр `-s`:

```
ln -s original.txt symlink.txt
```

3.3.2 Поводження з файлами та посиланнями

Видалення

Якщо видалити `original.txt`, то:

- `hardlink.txt` все ще буде повністю функціональним, оскільки він вказує на той самий *inode*.
- `symlink.txt` стане "битим"посиланням: воно залишиться, але вестиме до неіснуючого шляху.

Копіювання

Якщо скопіювати файл за допомогою `cp original.txt copy.txt`, створиться новий файл з новим *inode*, тобто копія є незалежною.

Якщо скопіювати символічне посилання без параметра `-P`, буде скопійовано саме посилання, а не файл:

```
cp symlink.txt new_symlink.txt
```

Якщо ж вказати `-L`, буде скопійовано файл, на який вказує symlink:

```
cp -L symlink.txt file_copy.txt
```

Переміщення

Якщо перемістити цільовий файл для символічного посилання, symlink стане недійсним.

Жорстке посилання залишиться функціональним до тих пір, поки зберігається *inode*.

3.3.3 Перевірка посилань

Щоб побачити, чи файл є символічним посиланням, і куди він вказує, можна використати команду:

```
ls -l
```

На екрані буде символ стрілки (->), що вказує на оригінальний файл.

Щоб побачити кількість жорстких посилань на файл:

```
ls -l
```

Другий стовець вказуватиме кількість посилань на цей *inode*.

3.3.4 Обмеження та відмінності

Hard link не можна створити для директорії (за винятком спеціальних випадків root-прав) або між різними файловими системами.

Symbolic link можна створити для директорій і між файловими системами.

Символічне посилання може бути абсолютним

```
/home/user/file.txt
```

або відносним (`../file.txt`).

3.3.5 Практичне застосування

Символічні посилання часто використовуються для створення дружніх імен для бібліотек або конфігураційних файлів. Наприклад:

```
ln -s /etc/nginx/sites-available/mysite
  /etc/nginx/sites-enabled/mysite
```

Жорсткі посилання іноді використовують для резервного копіювання даних без дублювання змісту.

Як бачимо, жорсткі та символічні посилання — важливі інструменти в Linux, які дозволяють гнучко керувати файлами і ресурсами. Розуміння їхньої структури і поведінки є необхідним для ефективної роботи з файловою системою.

Розділ 4

«Вікна» у світі комп'ютерів

4.1 Поява Microsoft. MS-DOS

Білл Гейтс із дитинства захоплювався програмуванням, і вже у школі створював перші власні програми. У 1975 році разом зі своїм другом *Полом Алленом* заснував компанію **Microsoft**, яка згодом стала однією з найбільших розробників програмного забезпечення.

Назва `Micro-soft` була утворена як поєднання слів «microcomputer» і «software». Спершу компанія розробляла інтерпретатори мови `BASIC` для ранніх персональних комп'ютерів, зокрема Altair 8800.

У 1980 році компанія IBM звернулася до Microsoft із запитом на створення операційної системи для їхнього нового персонального комп'ютера. Microsoft придбала існуючу ОС `86-DOS` у компанії Seattle Computer Products, адаптувала її та продала IBM під назвою **MS-DOS** (*Microsoft Disk Operating System*).

MS-DOS була *текстовою* операційною системою з ін-

терфейсом командного рядка. Вона стала стандартною ОС для більшості IBM-сумісних комп'ютерів протягом 1980-х років.

На відміну від сучасних багатозадачних ОС, MS-DOS була *однозадачною*, що означає — в один момент часу могла виконуватись лише одна програма.

Після завантаження системи користувач потрапляв у *командний інтерфейс*, де взаємодіяв із системою через введення команд у текстовому режимі. Користувач вводив команди після наступного запрошувального тексту:

```
C:\>
```

Деякі базові команди:

- `DIR` — вивести список файлів у директорії;
- `CD` — змінити директорію;
- `COPY`, `DEL`, `REN` — копіювати, видалити, перейменувати файл;
- `CLS` — очистити екран;
- `TYPE` — показати вміст текстового файлу;
- `EDIT` — простий текстовий редактор;
- `FORMAT`, `CHKDSK`, `FDISK` — робота з дисками.

Для запуску програми достатньо було ввести назву файлу, наприклад:

```
C:\PROGRAM.EXE
```

MS-DOS мала просту, модульну архітектуру:

- **IO.SYS** — низькорівнева взаємодія з пристроями введення/виведення.
- **MSDOS.SYS** — ядро операційної системи, що керувало файлами, пам'яттю, пристроями.
- **COMMAND.COM** — командний інтерпретатор, який виконував команди користувача.

При запуску комп'ютера BIOS завантажував IO.SYS та MSDOS.SYS з `boot-сектора`, після чого запускалась оболонка `COMMAND.COM`.

MS-DOS використовувала файлову систему **FAT12**, пізніше — **FAT16** (File Allocation Table). Основні її особливості:

- Максимальна довжина назви файлу — `8.3` (вісім символів і три символи розширення).
- Не підтримувалась ієрархічна структура (папки) понад 1 рівень до версії 2.0.
- Назви файлів були не чутливими до регістру.

Разом із тим, MS-DOS мала певні обмеження:

- *Пам'ять:* Підтримувалась лише **640 КБ** оперативної пам'яті. Додаткова (EMS/XMS) пам'ять потребувала окремих драйверів.
- *Багатозадачність:* Відсутня — можна було виконувати лише одне завдання за раз.
- *Безпека:* Відсутність будь-яких засобів захисту — всі програми працювали з повним доступом до системи.

- *Мережева підтримка:* Відсутня за замовчуванням; додавалась через сторонні драйвери.

Конфігурація MS-DOS здійснювалась через два ключові файли:

- `CONFIG.SYS` — завантаження драйверів та налаштування системних параметрів.
- `AUTOEXEC.BAT` — автоматичне виконання команд при запуску системи.

Приклад:

```
CONFIG.SYS:
DEVICE=HIMEM.SYS
DEVICE=EMM386.EXE

AUTOEXEC.BAT:
@ECHO OFF
PROMPT $P$G
PATH=C:\DOS;C:\UTILS
```

Попри обмеження, для MS-DOS було створено багато корисних програм:

- **Norton Commander** — двопанельний файловий менеджер.
- **Volkov Commander** — його легковісна альтернатива.
- **WordPerfect**, **Lotus 1-2-3** — текстовий редактор, електронні таблиці.
- **Turbo Pascal**, **Borland C++**, **QBasic** — популярні компілятори та середовища розробки.

MS-DOS стала основною платформою для ПК-ігор у 1980–90-х роках. Відомі ігри:

- **Prince of Persia**
- **Doom**, **Wolfenstein 3D**
- **SimCity**, **Civilization**
- **Warcraft**, **Dune II**

Ігри запускались із командного рядка, часто потребували налаштувань пам'яті, драйверів звукової карти (наприклад, `Sound Blaster`), конфігурацій файлів.

Microsoft завершила офіційну підтримку MS-DOS у 2000 році. Хоча DOS-команди залишалися частиною Windows іще деякий час, їхня функціональність була суттєво обмежена. У 64-бітних версіях Windows (починаючи з Windows XP x64) підтримка DOS-додатків повністю зникла.

Незважаючи на це, MS-DOS залишається важливим історичним етапом розвитку персональних комп'ютерів, а також предметом інтересу ентузіастів, ретро-ґеймерів і розробників низькорівневого програмного забезпечення.

Сьогодні DOS-програми можна запускати за допомогою емуляторів, таких як `DOSBox`, або через віртуальні машини. Це дозволяє відтворити середовище старих ПК, запускати класичні ігри чи демонструвати принципи роботи операційних систем студентам.

MS-DOS відіграла ключову роль у становленні комп'ютерної індустрії. Вона стала стартовою платформою для програмістів, ігор, офісного програмного забезпечення і самої Microsoft. Перехід до Windows як самостійної ОС символізував початок нової епохи — багатозадачних, графічних та мережевих операційних систем.

4.2 Розвиток графічного інтерфейсу: Windows

Після перших успіхів MS-DOS компанія Microsoft зрозуміла необхідність створення зручнішого, візуального інтерфейсу, що би дозволив працювати з комп'ютером не лише програмістам, а й звичайним користувачам. У 1985 році з'явилася перша графічна надбудова над MS-DOS — **Windows 1.0**. Її функціональність була обмеженою: вона дозволяла відкривати вікна, запускати кілька програм одночасно (в плитковому режимі), але не підтримувала перекривання одних вікон поверх інших. Windows 1.0 продавалась на дискетах, мала скромні вимоги, але вимагала заздалегідь встановленої MS-DOS. Сприйняття було стриманим — більшість користувачів не бачили великої користі в графіці, яка працювала повільно й виглядала примітивно.

У 1987 році вийшла **Windows 2.0**, яка вже дозволяла перекривати вікна, мала гарячі клавіші й кращу інтеграцію з прикладними програмами, такими як Excel та Word. Завдяки співпраці з IBM і поширенню ПК цієї марки, Windows поступово почала проникати в офіси. Хоча справжнього прориву ще не сталося, багато хто почав розглядати її як перспективну платформу.

Ситуація різко змінилася з появою **Windows 3.0** у 1990 році, а згодом — її оновлення **Windows 3.1** у 1992. Ці версії принесли істотні покращення в графіці, підтримку 256 кольорів, шрифтів TrueType, файлового менеджера та мультимедійних функцій. Windows 3.1 встановлювалася з дискет, коштувала близько $100, уперше ставши серйозною платформою для стороннього програмного забезпечення, включаючи ігри. У цей період Windows уже активно конкурувала з іншими системами, такими

як OS/2.

Із середини 1990-х років Microsoft поступово переходила від MS-DOS до графічного середовища **Windows** як до повноцінної операційної системи. Ранні версії Windows (1.0, 2.0, 3.x) працювали як графічні оболонки поверх MS-DOS, тобто потребували наявності DOS для запуску.

Ситуація змінилася з виходом **Windows 95**, яка хоч і містила внутрішню версію MS-DOS (7.0), але вже не потребувала його попереднього встановлення. Інтерфейс Windows став основним способом взаємодії з комп'ютером, а командний рядок — другорядним. Починаючи з **Windows 98** та особливо з **Windows ME**, DOS втрачав роль активного компонента системи.

Інсталяція MS-DOS зазвичай відбувалась із кількох дискет (наприклад, версія 6.22 — з трьох). Для інсталяції Windows 95, а згодом і Windows 98, використовувалися вже **CD-диски**, що значно спрощувало процес та дозволяло включати графічні інтерфейси, драйвери, мультимедійні компоненти та численні додаткові утиліти.

У 1995 році революційна **Windows 95** уперше об'єднала MS-DOS та графічний інтерфейс в єдину оболонку. Вона вперше запровадила меню «Пуск», панель завдань, і файлову систему FAT32. Хоча Windows 95 постачалася на CD-ROM, для оновлення можна було використовувати дискетну версію. Ціна продажу стартувала приблизно від $209 за повну версію. Для багатьох користувачів саме ця ОС стала першою Windows, яку вони побачили. Вона продавалася мільйонними накладами, а її вихід супроводжувався масштабною рекламною кампанією з піснею *Start Me Up* гурту Rolling Stones.

У 1998 році Microsoft випустила **Windows 98**, яка будувалась на основі 95, але мала покращену підтримку USB, графічні ефекти, взаємодію з мережею Інтер-

нет (наприклад, браузер Internet Explorer був частиною оболонки), і вперше запропонувала функцію автозавантаження з CD. Операційна система все ще залежала від MS-DOS, хоча й меншою мірою.

У 2000 році Microsoft представила **Windows Me (Millennium Edition)**, яка повинна була бути останньою DOS-залежною версією. Вона орієнтувалася на домашніх користувачів, мала підтримку мультимедіа, але зазнала значної критики через нестабільність і численні помилки.

Паралельно, в корпоративному сегменті Microsoft розвивала окрему лінійку — **Windows NT (New Technology)**. Перша версія з'явилась у 1993 році. Це була повністю 32-бітна ОС, що не залежала від DOS і мала модульну архітектуру з підтримкою багатозадачності, безпеки та мереж. MS-DOS більше не використовувався як основа. Він був лише *емуляційним середовищем* (через компонент `NTVDM` — NT Virtual DOS Machine), а вся архітектура системи базувалась на власному ядрі Windows NT.

У 2000 році з'явилася **Windows 2000**, яка об'єднала стабільність NT з деякими елементами зручності від Windows 98. Вона швидко стала популярною серед корпоративних користувачів.

У 2001 році вийшла **Windows XP**, яка вперше стала універсальною ОС для бізнесу та дому. Вона базувалась на ядрі NT, мала покращений інтерфейс Luna, систему оновлень через Інтернет, брандмауер і стабільну роботу. Windows XP продавалася як на CD, так і в комплекті з комп'ютерами. Її життєвий цикл був надзвичайно довгим: офіційна підтримка тривала до 2014 року. Ціна варіювалася — приблизно \$99–\$199 залежно від версії. XP стала культовою, завдяки великій кількості програм та ігор, сумісності й простоті.

У 2007 році Microsoft представила **Windows Vista** — з кардинально новим дизайном (Aero), покращеним захистом (User Account Control), та оновленою системою драйверів. Але вимоги до заліза (як-от 1 ГБ оперативної пам'яті) виявилися надто високими для багатьох тодішніх ПК, що призвело до негативного сприйняття.

Після Vista у 2009 році вийшла **Windows 7** — система, що врахувала всі помилки попередниці. Вона стала золотою серединою між функціональністю, стабільністю й продуктивністю. Ціна Windows 7 коливалась у межах \$120–\$200, і вона швидко стала найпопулярнішою ОС у світі, зберігаючи цей статус протягом багатьох років.

У 2012 році Microsoft спробувала змінити парадигму, випустивши **Windows 8** з акцентом на сенсорний інтерфейс і новий стартовий екран з «плитками». Проте відсутність звичного меню «Пуск» викликала хвилю критики. У 2013 році компанія випустила **Windows 8.1**, у якій частково виправила помилки, зокрема повернула кнопку «Пуск», яка все одно відкривала плитковий інтерфейс, проте багатьом користувачам система все одно не подобалась.

Ситуацію врятувала **Windows 10**, яка вийшла у 2015 році. Вона стала безкоштовною для користувачів Windows 7 і 8 протягом першого року після релізу. Microsoft уперше запровадила модель постійного оновлення — система отримувала великі функціональні оновлення декілька разів на рік. Windows 10 поєднувала класичний вигляд з новими функціями, мала високу продуктивність і стала основною платформою як для корпоративних користувачів, так і для ґеймерів.

У 2021 році з'явилася **Windows 11** — сучасна ОС із новим дизайном, централізованим меню «Пуск», інтеграцією з Android-додатками (через Amazon Appstore), під-

тримкою DirectStorage для ігор та TPM 2.0 як обов'язковою вимогою безпеки. Windows 11 поширювалася безкоштовно для користувачів Windows 10, однак вимоги до апаратного забезпечення суттєво зросли, що призвело до критики з боку власників старіших ПК.

Протягом десятиліть Windows залишалася найпоширенішою ОС на персональних комп'ютерах. Вона адаптувалася до потреб користувачів, змінюючи підходи до розповсюдження — від дискет і CD до цифрових завантажень з Microsoft Store, від коробкової моделі до хмарних ліцензій та OEM-поставок. Вона зазнавала як гучних провалів, так і безсумнівних успіхів, але завжди залишалась на передньому краї еволюції персональних систем.

4.3 Недоліки сучасних версій Windows та вибір на користь Linux

Попри значну еволюцію та поширеність, сучасні версії Windows мають ряд недоліків, які стають особливо помітними в корпоративному середовищі. По-перше, *вартість ліцензій* на Windows залишається суттєвою, особливо в масштабі великої організації. Ліцензії для підприємств або навчальних закладів передбачають окремі пакети (наприклад, Windows 11 Enterprise), ціна яких залежить від кількості пристроїв, функцій і потреб служби підтримки. Також часто потрібні платні ліцензії на додаткове програмне забезпечення Microsoft — такі як Office, Exchange або засоби керування мережею (наприклад, Microsoft Endpoint Manager).

По-друге, Windows часто критикують за *вразливості*

безпеки. Хоча система регулярно отримує оновлення, її поширеність робить її основною ціллю для шкідливого ПЗ: вірусів, троянів, програм-вимагачів. Стандарти безпеки постійно вдосконалюються, однак численні критичні оновлення свідчать про складність підтримки величезної кодової бази. Крім того, велика кількість програм сторонніх розробників для Windows збільшує ризики через слабке тестування чи несанкціоновані дії на рівні системи.

Ще одним недоліком є *високі системні вимоги*. Наприклад, Windows 11 вимагає наявність TPM 2.0, UEFI-завантаження та процесорів не нижче певних поколінь. Це виключає з оновлення мільйони пристроїв, які могли б ще роками працювати з менш вимогливими ОС. Також сама система займає десятки гігабайт на диску, має складну структуру автозапуску, системних служб і залежностей.

Окрему критику викликає система *примусових оновлень*. Користувачі не завжди можуть повністю контролювати, коли і як оновлення встановлюються. В деяких випадках після оновлень система може працювати нестабільно, втрачати сумісність із драйверами або стороннім програмним забезпеченням.

На тлі цих проблем дедалі більше користувачів — особливо в IT-інфраструктурі підприємств — звертають увагу на *Linux*. Ця система має низку переваг:

- Вона *безкоштовна* і поширюється за відкритою ліцензією (`GPL` та інші).

- Ядро Linux та багато дистрибутивів є *відкритими*, що дозволяє перевірити код на наявність бекдорів або вразливостей.

- Linux має гнучку *архітектуру доступів*, потужну систему прав (`chmod`, `sudo`), чіткий розподіл між користувачем та адміністратором.

- Система *легша й швидша*: більшість дистрибутивів можуть працювати на старому обладнанні, запускатися з USB або бути повністю керованими з командного рядка.

- *Оновлення* в Linux повністю контролюються адміністратором, часто не потребують перезавантаження і охоплюють усю систему, включаючи стороннє програмне забезпечення.

У 2017 році ці переваги стали особливо помітними під час розповсюдження шкідливого програмного забезпечення **Petya/NotPetya**, яке вразило десятки державних установ та підприємств в Україні. Вірус використовував вразливості у протоколі SMB, а також механізми MBR, які характерні саме для Windows. У багатьох випадках Petya шифрував жорсткі диски та блокував завантаження системи. Поширення відбувалося через заражені оновлення бухгалтерського ПЗ або відкриті порти у внутрішній мережі.

Linux-сервери та робочі станції виявилися практично невразливими, оскільки механізмів, якими користувався Petya (зокрема, специфіка реалізації MBR і API Windows), просто не існувало в Linux. Крім того, адміністратори Linux-систем зазвичай не запускають під обліковим записом `root` сторонні програми, а структура дистрибутивів не дозволяє виконати шкідливий код без відповідних прав.

Таким чином, безкоштовність, прозорість, безпека та гнучкість Linux роблять її привабливою альтернативою

для багатьох сфер — від серверного хостингу до офісних систем та навчальних закладів.

Розділ 5

«Яблука» за ціною діамантів

5.1 Поява і розвиток MacOS

Компанія **Apple Inc.** є однією з найвідоміших технологічних компаній у світі, що відіграла ключову роль у формуванні персональних комп'ютерів, мобільних пристроїв та операційних систем. Історія Apple почалася у 1976 році, коли *Стів Джобс*, *Стів Возняк* і *Рон Вейн* заснували компанію в гаражі батьків Джобса у Лос-Альтосі, Каліфорнія. Першим продуктом став комп'ютер **Apple I** — плата без корпусу, клавіатури та монітора, зібрана вручну Возняком.

Успіх прийшов із появою **Apple II** — одного з перших масових персональних комп'ютерів з кольоровою графікою, який став надзвичайно популярним у школах, офісах і домашньому користуванні. Згодом Apple намагалася зробити прорив із проєктом **Lisa**, який став першим комп'ютером з графічним інтерфейсом користувача (GUI) і мишкою. Проте надвисока ціна зробила його

комерційно невдалим.

У 1984 році компанія представила **Macintosh** — відносно доступний комп'ютер з графічним інтерфейсом, натхненним розробками Xerox PARC. Macintosh був початком цілого покоління комп'ютерів Apple з власною операційною системою **Mac OS**.

У той час Стів Джобс запросив до компанії керівника Pepsi **Джона Скаллі**, заманивши його крилатою фразою: *«Ти хочеш усе життя продавати солодку воду — чи хочеш змінити світ?»* Попри початкові успіхи, між Джобсом і Скаллі зростала напруга. У 1985 році рада директорів фактично усунула Джобса з компанії через невдоволення його управлінськими рішеннями.

Після цього Джобс заснував нову компанію **NeXT Inc.**, яка створила потужну, але дорогу робочу станцію NeXT і операційну систему NeXTSTEP — UNIX-подібну систему з об'єктно-орієнтованим інтерфейсом. Саме ця система згодом стане основою для майбутньої macOS.

Apple переживала кризу в 90-х роках. Різноманітні моделі комп'ютерів, слабке програмне забезпечення та відсутність інновацій знижували популярність компанії. У 1997 році Apple викупила компанію NeXT і повернула Стіва Джобса, який незабаром став тимчасовим, а потім постійним генеральним директором.

На основі NeXTSTEP з'явилася нова операційна система **Mac OS X**, що використовувала ядро **Darwin** — відкриту реалізацію на базі **BSD UNIX**. До цього Apple використовувала класичну **Mac OS** (від версії System 1 до 9), яка не мала сучасних механізмів пам'яті та багатозадачності. З OS X система отримала захист пам'яті, справжню багатозадачність, командний рядок і стабільне UNIX-середовище.

У 2000-х роках під керівництвом Джобса Apple поча-

ла запуск портативних продуктів: **iPod**, **iPhone**, **iPad**. Запуск онлайн-магазину додатків **App Store** кардинально змінив спосіб дистрибуції програмного забезпечення. Незалежні розробники отримали змогу створювати власні додатки й розповсюджувати їх. Комп'ютери iMac та ноутбуки MacBook отримували новий дизайн.

Після смерті Стіва Джобса у 2011 році компанію очолив **Тім Кук**. Під його керівництвом Apple продовжила зростати, зосередившись на стабільності, екосистемі пристроїв і сервісах (таких як iCloud, Apple Music, Apple Pay). Продуктова лінійка розширилася на носимі пристрої (**Apple Watch**, **AirPods**), а також процесори власного виробництва (**Apple Silicon** — M1, M2, M3).

5.2 Сучасні версії macOS

Сучасна операційна система macOS ґрунтується на багаторівневій модульній архітектурі, побудованій на відкритому ядрі *XNU* (скорочення від *X is Not Unix*), яке поєднує елементи мікроядра *Mach* від Carnegie Mellon University та компонентів *FreeBSD*, що забезпечують POSIX-сумісність. У підґрунті macOS лежить *Darwin* — Unix-подібна система, яка відкриває доступ до командного рядка та можливостей для розробників, тоді як графічне середовище забезпечує інтуїтивний інтерфейс користувача з використанням Quartz-композитного рушія.

У контексті безпеки система реалізує багатокомпонентну модель захисту. Розмежування прав між користувачами та додатками відбувається завдяки механізмам *System Integrity Protection* (SIP), *Gatekeeper*, а також підписанню коду, що дозволяє запуск лише перевірених програм. Крім того, macOS підтримує *sandboxing* для ізоляції процесів і шифрування файлової системи за допомогою

FileVault 2.

Важливо зазначити, що macOS є 64-розрядною системою і вже давно не підтримує 32-бітні програми. Вона оптимізована для роботи з фірмовими ARM-чіпами *Apple Silicon* (лінійка M1, M2, M3), хоча певний період зберігала підтримку x86_64-архітектури на пристроях із процесорами Intel. MacOS має гнучку систему керування пам'яттю, включно з віртуалізацією, кешуванням та динамічним керуванням ресурсами, що забезпечує багатозадачність навіть у ресурсомістких середовищах.

macOS X 10.0 "Cheetah" була першим релізом, хоча ще недопрацьованим і повільним. Версія 10.1 "Puma" значно покращила швидкодію та стабільність. Подальші версії — "Jaguar" (10.2), "Panther" (10.3) і особливо "Tiger" (10.4) — поступово перетворили систему на повноцінне середовище для роботи, з підтримкою сучасних технологій, таких як Spotlight, Dashboard, iChat AV та оновленого Safari.

macOS 10.5 "Leopard" і 10.6 "Snow Leopard" внесли зміни на рівні інтерфейсу, 64-бітної архітектури та підвищення стабільності. Snow Leopard вважається однією з найуспішніших версій, особливо серед професіоналів. Наступні оновлення — "Lion" (10.7) та "Mountain Lion" (10.8) — зблизили macOS з iOS, додавши Launchpad, Notification Center та інші спільні елементи.

У 10.9 "Mavericks" Apple відмовилась від назви великих котів, почавши використовувати географічні назви Каліфорнії. Версії "Yosemite" (10.10) та "El Capitan" (10.11) принесли редизайн у стилі flat UI та вдосконалення в роботі з продуктивністю.

У 2016 році з виходом macOS 10.12 "Sierra" Apple офіційно змінила назву на *macOS*, підкреслюючи єдність лінійки iOS, tvOS та watchOS. Наступні релізи включали "High Sierra", "Mojave" з темною темою, "Catalina", яка

відмовилась від підтримки 32-бітних програм, “Big Sur” з великим редизайном та переходом на чипи Apple Silicon, “Monterey”, “Ventura” та найновішу версію “Sonoma”.

Зміна архітектури з Intel на ARM (Apple M1, M2) стала найважливішим кроком у розвитку macOS, що забезпечив вищу енергоефективність, безшумність та довшу автономність ноутбуків Mac. Усі сучасні версії macOS підтримують оновлення через Mac App Store, мають інтеграцію з iCloud, iPhone та іншими пристроями Apple, і надають користувачам розвинену графічну оболонку та високу стабільність, хоча зберігають закриту модель поширення та ліцензування.

Особливістю екосистеми Apple є її закритість і сувора інтеграція з апаратним забезпеченням. Це дозволяє домогтися певної стабільності та продуктивності, але обмежує користувача в можливостях налаштування та модифікації системи. З іншого боку, така інтеграція забезпечує взаємодію між пристроями Apple, які об’єднуються в єдине середовище через Handoff, iCloud, AirDrop тощо.

Система macOS не є безкоштовною у загальному розумінні: вона офіційно розповсюджується лише з фірмовими пристроями Apple, ціна яких є високою. І хоча сама ОС є безкоштовною, вартість входу в екосистему є суттєвою. Крім того, macOS часто критикують за складність персоналізації, обмеження при встановленні стороннього ПЗ та невелику кількість вільного програмного забезпечення порівняно з Linux. Платформа macOS здебільшого орієнтована на замкнутий набір власного ПЗ.

Прив’язка macOS виключно до обладнання Apple робить її *недоступною* для користувачів сторонніх ПК — офіційно вона не підтримує встановлення на інші системи (хоча існує спільнота *Hackintosh*, яка обходить це обмеження). Також macOS не має стільки програмного забезпе-

чення для ігор або спеціалізованого ПЗ як Windows.

На відміну від Windows, macOS є більш захищеною і менш схильною до вірусів, однак вона поступається **Linux** у гнучкості, налаштуванні, і зокрема у серверних чи наукових застосуваннях. Багато розробників і організацій надають перевагу Linux завдяки його відкритості, контролю над оновленнями, відсутності прив'язки до бренду та спільноті з відкритим кодом.

Розділ 6

Комп'ютер у кишені

6.1 Мобільні операційні системи на початку тисячоліття

Історія мобільних операційних систем бере свій початок у 1990-х роках, коли мобільні телефони поступово еволюціонували від простих пристроїв для дзвінків і SMS до багатофункціональних кишенькових комп'ютерів. Однією з перших справжніх платформ стала *Symbian OS*, яка розроблялася консорціумом кількох компаній, зокрема Nokia.

Symbian була однією з найважливіших мобільних операційних систем початку 2000-х років і справедливо вважається піонером у сфері смартпристроїв. Система виникла в результаті трансформації платформи EPOC, розробленої компанією Psion, і згодом розвивалась як спільний проєкт кількох провідних виробників мобільної техніки, зокрема Nokia, Ericsson, Motorola та Sony Ericsson, які сформували консорціум *Symbian Ltd.*.

На архітектурному рівні Symbian OS була побудована як *мікроядрова* система з чітким розділенням між ядром,

сервісами та інтерфейсами. Це забезпечувало модульність, гнучкість і високу стабільність. Основна модель виконання базувалась на використанні *active objects* замість традиційних потоків, що дозволяло ефективно працювати з обмеженими ресурсами мобільних пристроїв того часу. Завдяки асинхронному підходу до обробки подій Symbian мала низьке енергоспоживання та швидкий відгук інтерфейсу.

Однією з визначальних рис Symbian була її *реальновіріантна багатозадачність* із захистом пам'яті, що дозволяло одночасно працювати з кількома додатками без взаємного впливу або збоїв. Система підтримувала різноманітні рівні API, у тому числі *Symbian C++*, який надавав доступ до низькорівневих можливостей пристрою. Також пізніше було додано підтримку Java ME, що значно розширило коло доступних додатків.

Symbian OS використовувалась на широкій гамі пристроїв, зокрема таких моделей як `Nokia 6600`, `Nokia N70`, `Nokia E71`, `Nokia N95`, `Sony Ericsson P1i`, `Samsung i8910 HD` та багатьох інших. З часом Symbian розвивалася у кілька варіантів інтерфейсів, серед яких найпопулярнішими були *Series 60* (S60), *UIQ* та *MOAP(S)*. Платформа S60, розроблена Nokia, стала стандартом де-факто серед смартфонів того періоду й вирізнялася дружнім інтерфейсом, широкими мультимедійними можливостями та підтримкою бездротових технологій (Bluetooth, Wi-Fi, 3G).

Symbian підтримувала складні функції, як-от багатокамерна зйомка, GPS-навігація, мобільна пошта, Інтернет-браузери, синхронізація з ПК, використання SD-карт та FM-радіо, що було новаторським для початку 2000-х років. Додатки для Symbian встановлювалися з `.sis` або `.sisx` пакетів, які могли підписуватись для забезпечення

безпеки.

Symbian залишила глибокий слід в історії мобільних технологій. Це була система, що вперше реалізувала у широкому масштабі концепцію *смартфона* — пристрою, здатного поєднувати в собі функції телефону, КПК, камери, медіаплеєра та інтернет-комунікатора. Її вплив на подальший розвиток мобільних ОС є беззаперечним.

У 2010 році Nokia спробувала модернізувати Symbian через версію *Symbian 3*, яка принесла підтримку мультитач-інтерфейсу та нову графічну підсистему, однак ринок уже поступово зміщувався до платформ нового покоління. Тим не менш, Symbian залишалася функціональною і популярною системою ще протягом кількох років, особливо в регіонах, де цінували надійність, автономність та широкий набір функцій без надмірної вартості пристрою. Таким чином, Symbian була віховою технологією у світі мобільних ОС. Вона не лише відкрила епоху смартфонів, але й задала багато архітектурних принципів, які згодом перейняли інші платформи.

Паралельно з Symbian на ринку з'явилася платформа *BlackBerry OS* від компанії Research In Motion. Вона була популярною серед корпоративних користувачів завдяки вбудованим засобам захисту, шифруванню повідомлень та апаратній клавіатурі. Проте обмежена екосистема, відсутність зручного магазину додатків і слабка адаптація до сенсорних інтерфейсів зробили BlackBerry неконкурентною в епоху смартфонів.

На початку 2000-х Microsoft представила *Windows CE*, а згодом *Windows Mobile*, орієнтовані на КПК (кишенькові персональні комп'ютери) та бізнес-пристрої. Вони пропонували підтримку стилуса, багатозадачність і навіть спрощені версії офісних програм, однак громіздкий інтерфейс, часті збої та несумісність між версіями ста-

ли серйозною перешкодою для масового прийняття. Пізніше спроба перезапустити мобільний напрям у вигляді *Windows Phone* з новим інтерфейсом Metro і оновленим ядром не знайшла значної підтримки серед користувачів та розробників. Обмежена кількість додатків у магазині Windows Store, повільне оновлення пристроїв і погана маркетингова стратегія призвели до повного згортання мобільної платформи Microsoft.

6.2 Сучасний ринок

У 2007 році компанія Apple представила iPhone з операційною системою *iOS*. Вона стала першою мобільною платформою з повністю сенсорним інтерфейсом, простим і інтуїтивно зрозумілим дизайном та централізованим магазином App Store, який створив нову екосистему для розробників і бізнесу. iOS швидко здобула популярність серед споживачів завдяки стабільності, швидкості, регулярним оновленням та високому рівню безпеки. Однак закритість системи, обмеження у налаштуваннях та висока вартість пристроїв залишаються предметом критики й до сьогодні.

Майже одночасно компанія Google випустила *Android*, мобільну операційну систему з відкритим кодом, побудовану на ядрі *Linux*. Android від самого початку позиціювався як гнучка платформа, яку можна було адаптувати до різних пристроїв. Завдяки відкритості, великій кількості виробників (Samsung, HTC, LG, Huawei та інші) та підтримці з боку Google, Android став найпопулярнішою мобільною ОС у світі. Google Play, офіційний магазин додатків, поступово перетворився на найбільшу платформу для мобільного ПЗ, хоча й не позбавлений проблем із модерацією та шкідливим вмістом. Android цінується

за гнучкість, широку підтримку обладнання, різноманіття цінових сегментів, проте страждає від фрагментації — велика кількість пристроїв мають різні версії ОС, що ускладнює оновлення й забезпечення безпеки.

Операційна система Android має модульну багаторівневу архітектуру, що базується на модифікованому ядрі *Linux*. На найнижчому рівні знаходиться ядро, яке відповідає за керування пам'яттю, процесами, драйверами пристроїв та безпекою. Поверх нього розташований рівень *HAL* (Hardware Abstraction Layer), що надає уніфікований інтерфейс до апаратного забезпечення. Наступним шаром є бібліотеки та середовище виконання — *Android Runtime (ART)*, яке виконує байт-код програм, написаних на Java або Kotlin, попередньо перетворених у формат `.dex`. Вище розташовані фреймворки, які надають розробникам інтерфейси для доступу до системних сервісів (керування вікнами, сповіщення, дані, зв'язок). На верхньому рівні — користувацькі застосунки. Важливо зазначити, що Android підтримує модель ізольованих додатків, кожен з яких працює у власному процесі та середовищі виконання, з обмеженим доступом до системних ресурсів, що підвищує безпеку.

У свою чергу, архітектура iOS побудована на базі *Darwin*, що є UNIX-подібним ядром, створеним компанією Apple. Darwin включає в себе ядро *XNU*, яке поєднує в собі мікроядро Mach та компоненти BSD. Поверх ядра працює шар низькорівневих системних бібліотек, зокрема POSIX-сумісні API. Основні функціональні можливості забезпечуються фреймворками, такими як *Core Foundation*, *Core Graphics*, *Core Data*, *UIKit* тощо. Для розробки додатків використовується мова Swift або Objective-C. Кожен додаток запускається в окремому "пісочному середовищі"(*sandbox*), що суворо обмежує доступ до си-

стемних ресурсів і даних інших додатків. Це забезпечує високий рівень безпеки, хоча й накладає обмеження на взаємодію між застосунками.

Обидві операційні системи мають власні магазини додатків — *Google Play* для Android та *App Store* для iOS, які реалізують політику перевірки, підпису та контролю поширення програмного забезпечення. Android є відкритою системою з відкритим вихідним кодом (проект AOSP), що дозволяє виробникам змінювати інтерфейс і додавати власні компоненти, тоді як iOS залишається повністю закритою системою, керованою компанією Apple.

Екосистеми iOS та Android продовжують еволюціонувати. Google просуває Android у різні сфери — телефони, телевізори, автомобілі (Android Auto), носимі пристрої (Wear OS) та навіть ноутбуки (Chrome OS). Apple, у свою чергу, інтегрує свої пристрої в єдину систему, що охоплює iPhone, iPad, Mac, Apple Watch і Apple TV, забезпечуючи плавний обмін даними, дзвінки, повідомлення, нотатки, синхронізацію програм.

Сучасні мобільні ОС, попри всю свою потужність, мають і недоліки. iOS критикують за закритість, високу вартість додатків, обмежений контроль над системою. Android — за нерівномірну якість реалізації на різних пристроях, затримки з оновленнями, складність забезпечення безпеки на старих моделях. Водночас обидві платформи демонструють зрілість, мають розвинені API, безпечні магазини додатків і підтримують мільярди користувачів по всьому світу.

Загалом, мобільні операційні системи пройшли довгий шлях — від складних і незручних платформ до високотехнологічних середовищ, які замінили комп'ютери у повсякденному житті мільйонів людей.

Розділ 7

Лабораторні роботи

7.1 Лабораторна робота №1. Встановлення Ubuntu

1. Встановити програму VirtualBox, якщо її немає. Завантажити її можна на офіційному сайті. Створити в ній нову віртуальну машину. При створенні вказати обсяг оперативної пам'яті (RAM) 2048 МБ або більше. Жорсткий диск – 30 ГБ.

2. Завантажити операційну систему Ubuntu. Це файл у форматі ISO, який займає близько 2 ГБ.

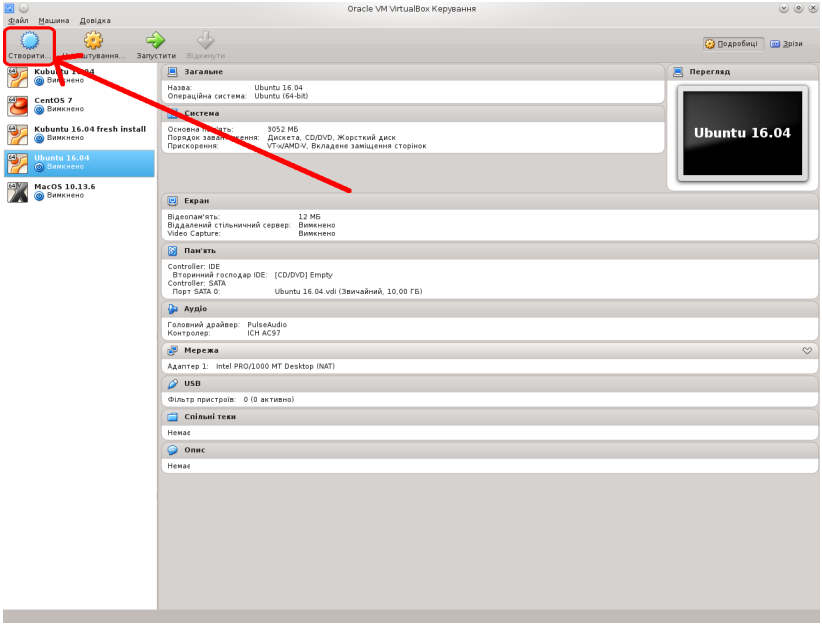
    - Якщо у вас 64-розрядний процесор, то завантажити найновішу версію Ubuntu.
    - Якщо вона підвисає або взагалі не працює, то можна завантажити старішу Ubuntu 20.04 або 18.04 (вибрати Desktop image -> 64-bit PC (AMD64) desktop image).
    - Якщо ж у вас 32-розрядний процесор, то остання версія, яка підтримує їх, – Ubuntu 16.04, її

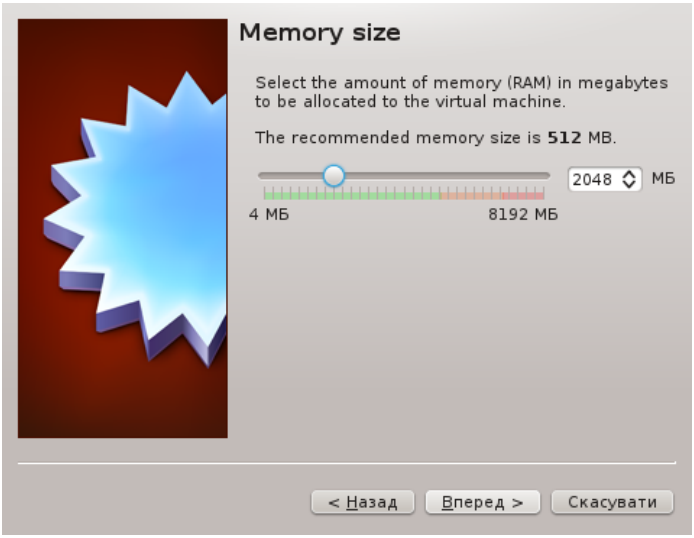
можна завантажити на офіційному сайті Ubuntu (вибрати Desktop image -> 32-bit PC (i386) desktop image).

3. Після встановлення Ubuntu зайти до системи і завантажити в магазині додатків Ubuntu Software Center три довільні програми на власний вибір (але не Viber, Telegram чи Spotify).

7.1.1 Як встановити Ubuntu на віртуальній машині

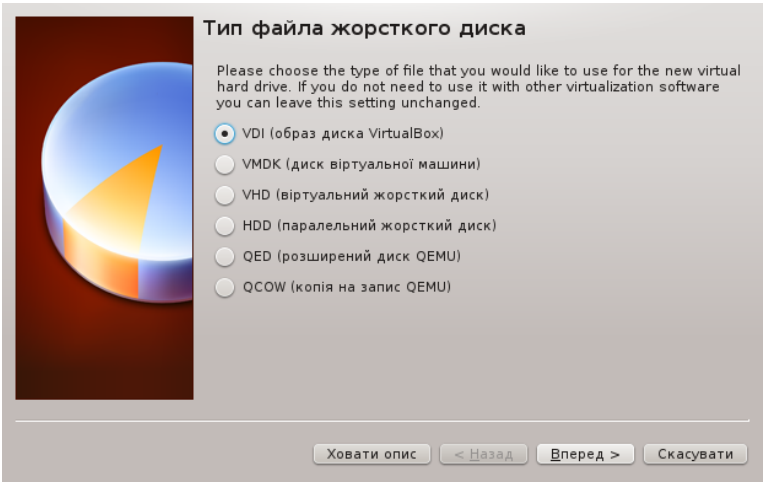
Запускаємо програму VirtualBox і натискаємо "Створити щоб створити новий віртуальний комп'ютер (машину).



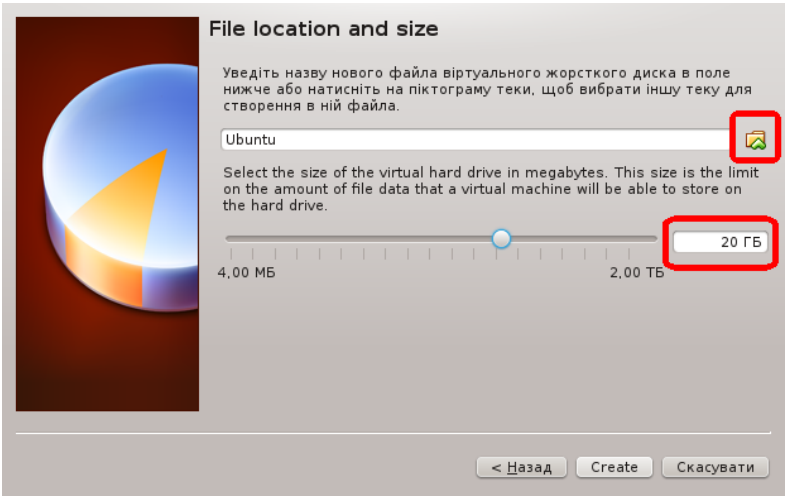


Оперативної пам'яті виділяємо не менше 2 ГБ (тобто 2048 МБ).

Після цього виділяємо частину жортского диска.

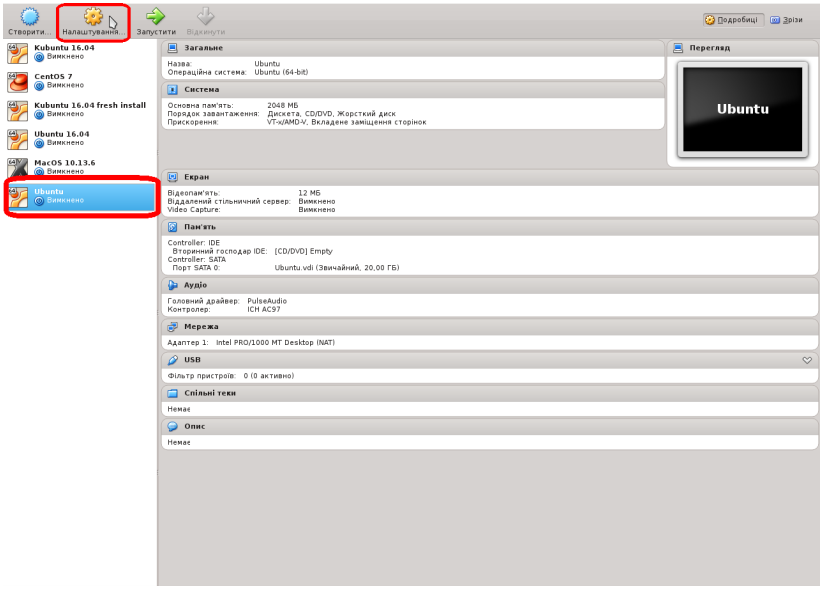


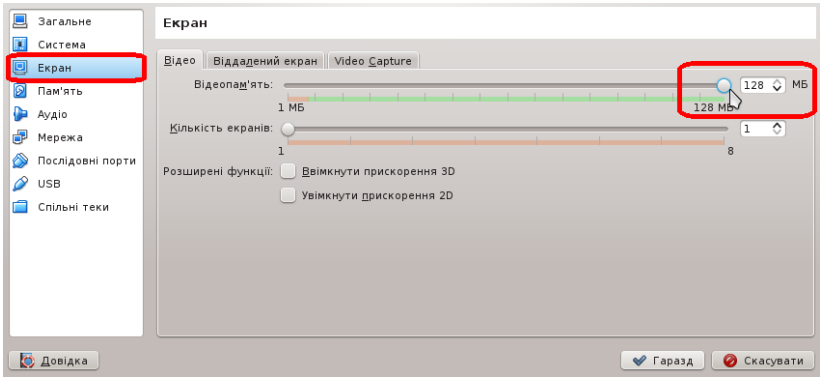
Далі потрібно вказати, де будуть зберігатись файли віртуальної машини. Рекомендується обирати диск D, а не C. Переконайтесь, що там є 30 ГБ вільного місця (для старих версій Ubuntu достатньо 20, але для нових версій краще 30).



Після цього нова віртуальна машина з'явиться у списку ліворуч. Перед запуском треба задати для неї ще деякі налаштування.

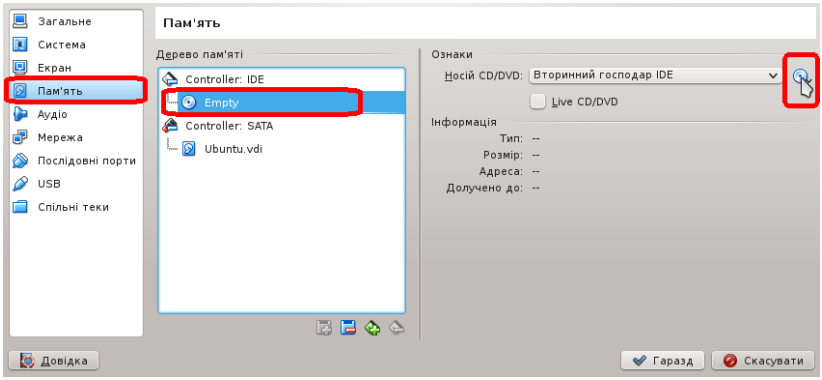
Для цього натиснемо кнопку «Налаштування» вгорі вікна.

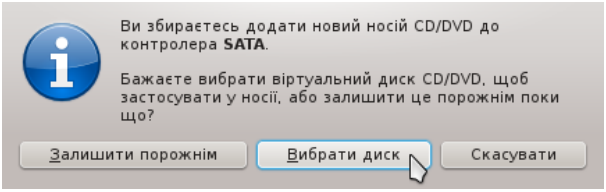


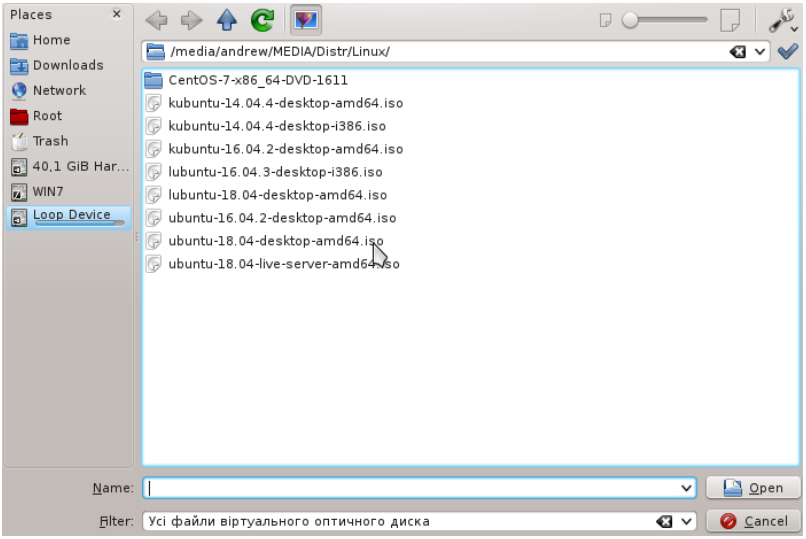


Для коректної роботи віртуальної машини задамо не менше 128 МБ відеопам'яті.

Наступним етапом треба вказати віртуальній машині, де знаходиться ISO-файл Ubuntu.

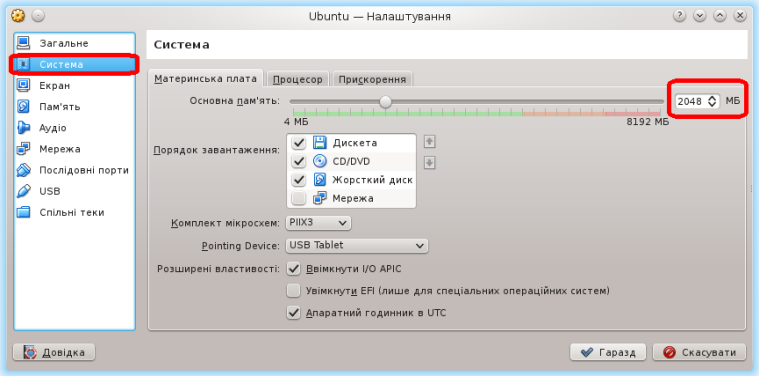

Загальне
Система
Екран
Пам'ять
Аудіо
Мережа
Послідовні порти
USB
Спільні теки
Пам'ять
Дерево пам'яті
Controller: IDE
Empty
Controller: SATA
Ubuntu.vdi
Ознаки
Носій CD/DVD:
Вторинний господар IDE
Live CD/DVD
Інформація
Тип: --
Розмір: --
Адреса: --
Долучено до: --
Довідка
Гаразд
Скасувати

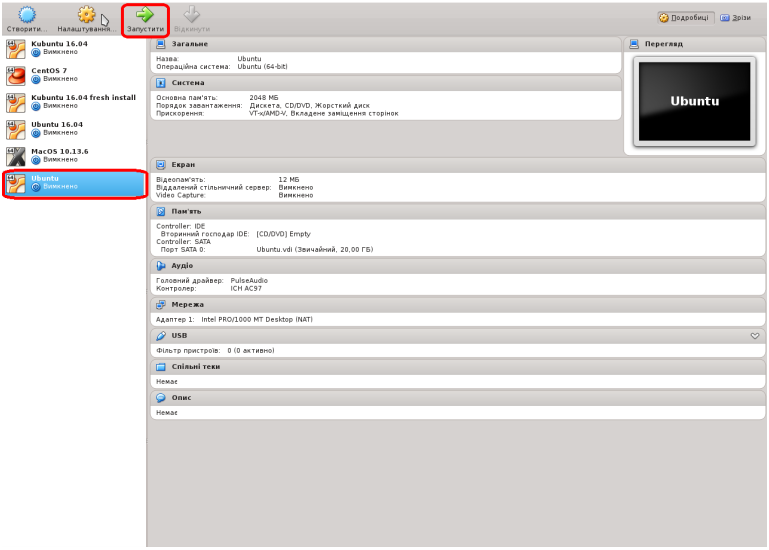

Ви збираєтесь додати новий носій CD/DVD до контролера SATA.
Бажаєте вибрати віртуальний диск CD/DVD, щоб застосувати у носії, або залишити це порожнім поки що?
Залишити порожнім
Вибрати диск
Скасувати


Places
Home
Downloads
Network
Root
Trash
40,1 GiB Har...
WIN7
Loop Device
/media/andrew/MEDIA/Distr/Linux/
CentOS-7-x86_64-DVD-1611
kubuntu-14.04.4-desktop-amd64.iso
kubuntu-14.04.4-desktop-i386.iso
kubuntu-16.04.2-desktop-amd64.iso
lubuntu-16.04.3-desktop-i386.iso
lubuntu-18.04-desktop-amd64.iso
ubuntu-16.04.2-desktop-amd64.iso
ubuntu-18.04-desktop-amd64.iso
ubuntu-18.04-live-server-amd64.iso
Name:
Open
Filter:
Усі файли віртуального оптичного диска
Cancel

Він буде слугувати завантажувальним диском, який запустить машина одразу при запуску, оскільки на її віртуальному жорсткому диску ще нічого не встановлено.

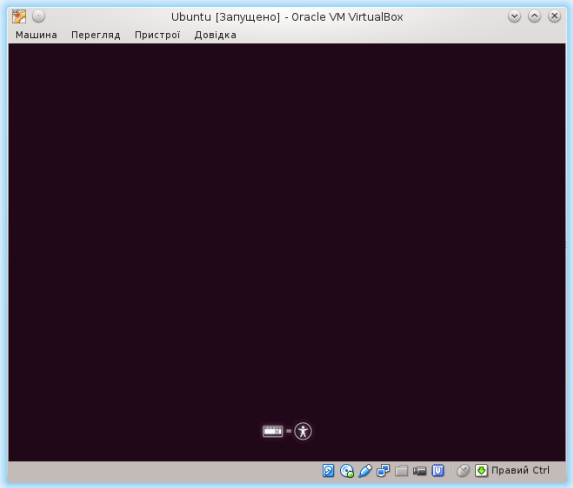
Перевіримо налаштування оперативної пам'яті.

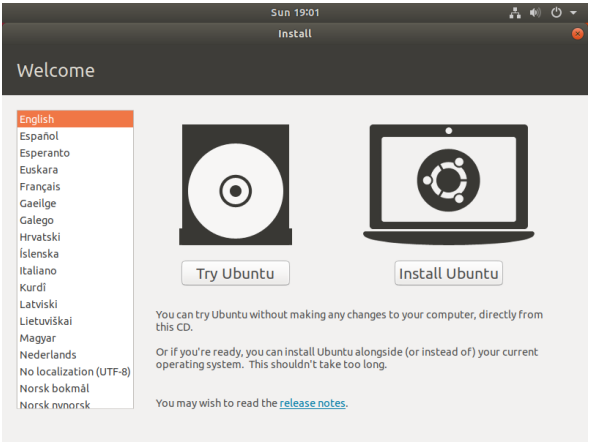


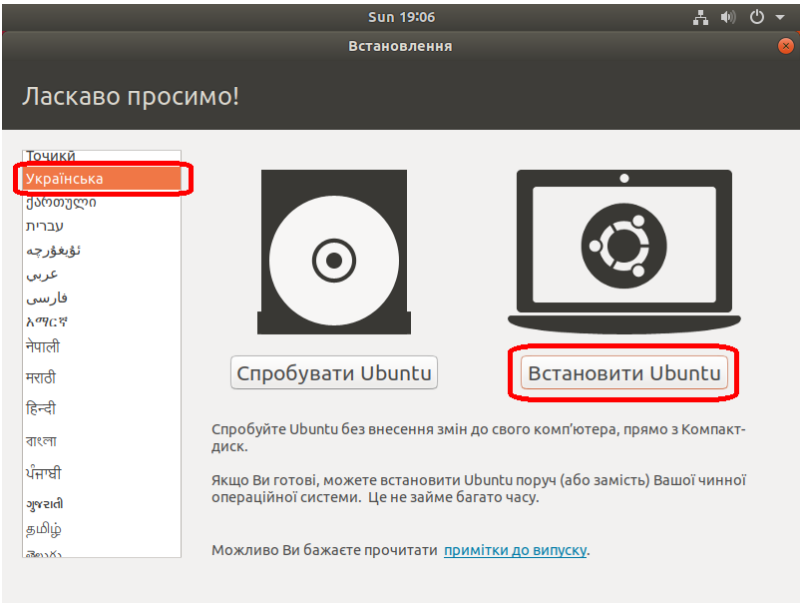


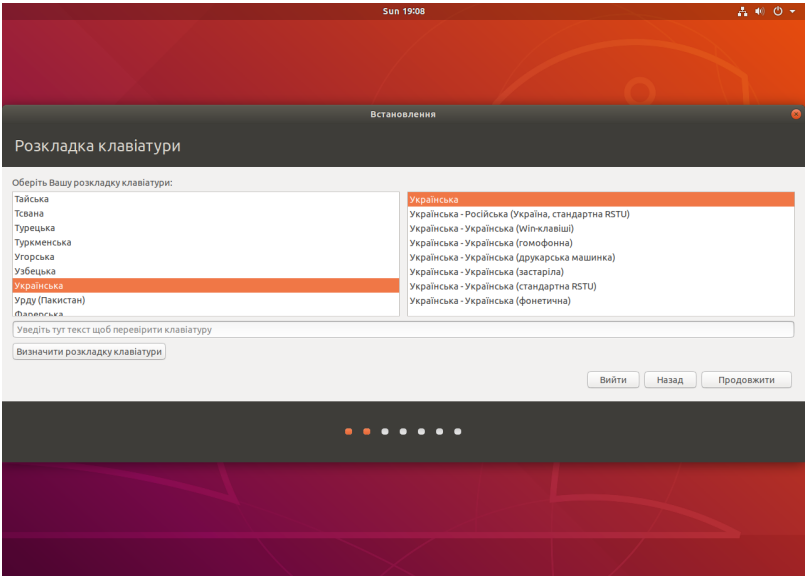
Після всіх налаштувань натискаємо «Гаразд». Далі натискаємо кнопку «Запустити», щоб запустити новостворену віртуальну машину.

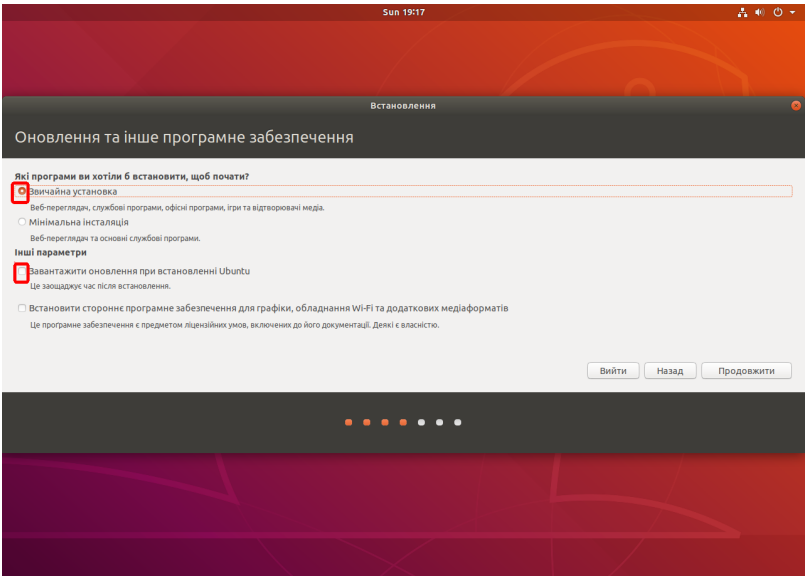
Якщо все налаштовано правильно, то має запуститися ось таке вікно:

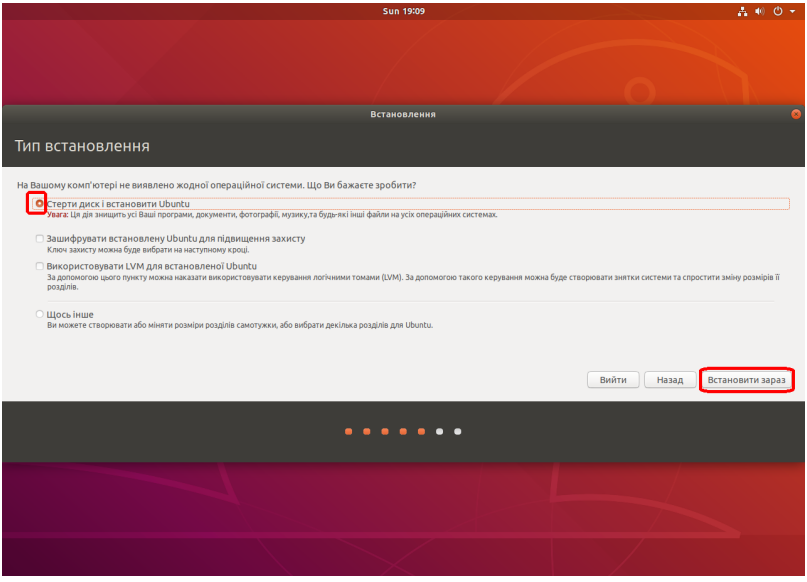


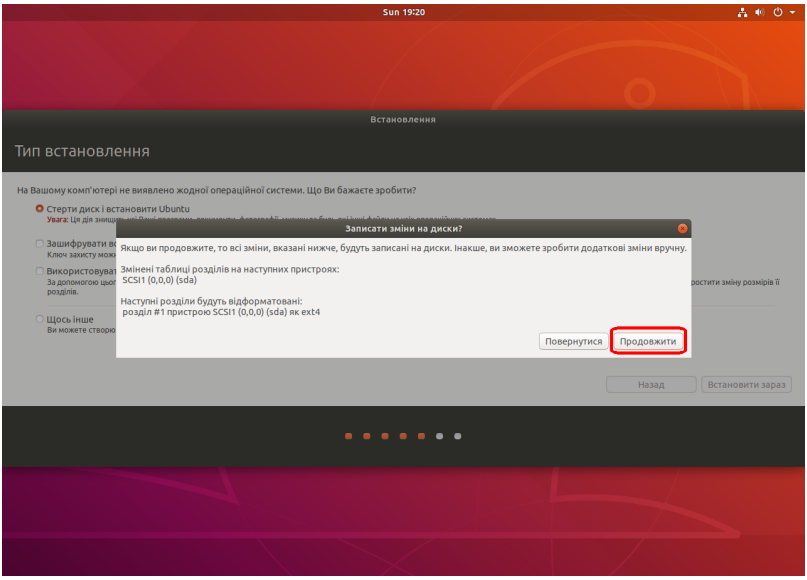


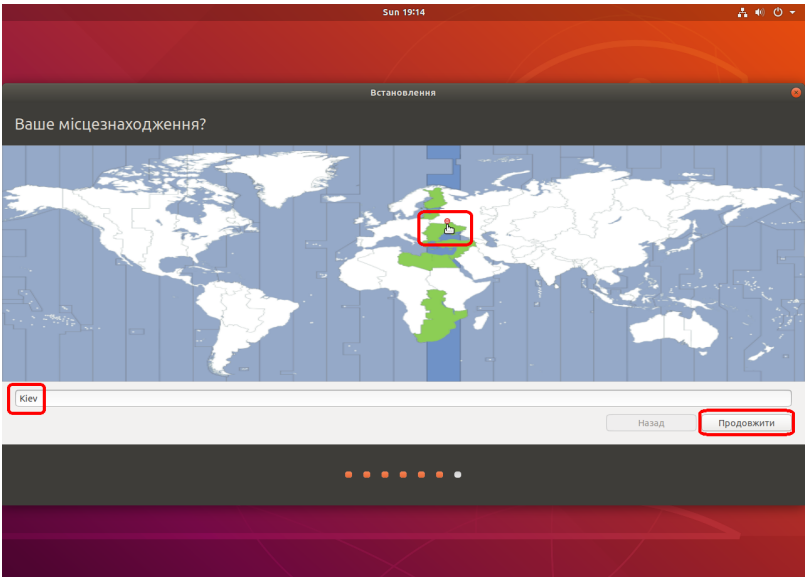

Sun 19:06
Встановлення
Ласкаво просимо!
Точикй
Українська
ქართული
עברית
ئۇيغۇرچە
عربی
فارسی
አማርኛ
नेपाली
मराठी
हिन्दी
বাংলা
ਪੰਜਾਬੀ
ગુજરાતી
தமிழ்
Спробувати Ubuntu
Встановити Ubuntu
Спробуйте Ubuntu без внесення змін до свого комп'ютера, прямо з Компакт-диск.
Якщо Ви готові, можете встановити Ubuntu поруч (або замість) Вашої чинної операційної системи. Це не займе багато часу.
Можливо Ви бажаєте прочитати примітки до випуску.

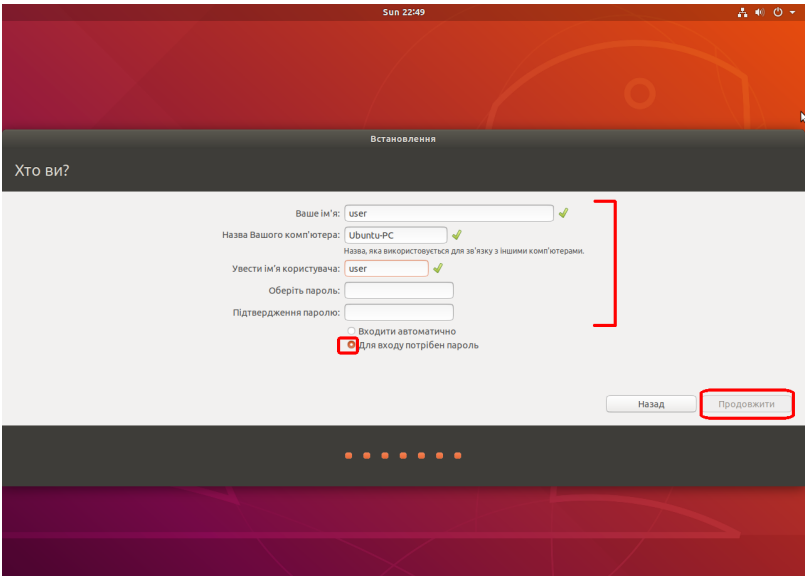

Sun 19:08
Встановлення
Розкладка клавіатури
Оберіть Вашу розкладку клавіатури:
Тайська
Тсвана
Турецька
Туркменська
Угорська
Узбецька
Українська
Урду (Пакистан)
Українська
Українська - Російська (Україна, стандартна RSTU)
Українська - Українська (Win-клавіші)
Українська - Українська (гомофонна)
Українська - Українська (друкарська машинка)
Українська - Українська (застаріла)
Українська - Українська (стандартна RSTU)
Українська - Українська (фонетична)
Уведіть тут текст щоб перевірити клавіатуру
Визначити розкладку клавіатури
Вийти
Назад
Продовжити

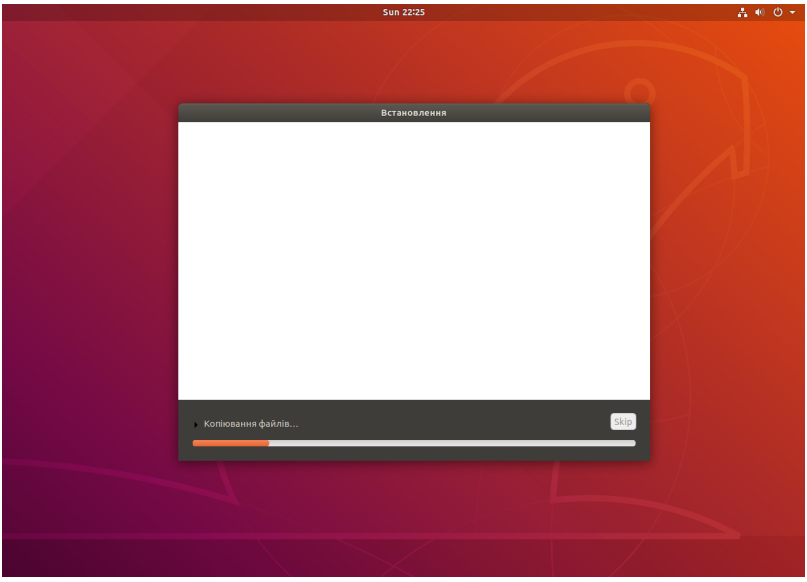

Sun 19:17
Встановлення
Оновлення та інше програмне забезпечення
Які програми ви хотіли б встановити, щоб почати?
Звичайна установка
Веб-переглядач, службові програми, офісні програми, ігри та відтворювачі медіа.
Мінімальна інсталяція
Веб-переглядач та основні службові програми.
Інші параметри
Завантажити оновлення при встановленні Ubuntu
Це заощаджує час після встановлення.
Встановити стороннє програмне забезпечення для графіки, обладнання Wi-Fi та додаткових медіаформатів
Це програмне забезпечення є предметом ліцензійних умов, включених до його документації. Деякі є власністю.
Вийти
Назад
Продовжити


Sun 19:09
Встановлення
Тип встановлення
На Вашому комп'ютері не виявлено жодної операційної системи. Що Ви бажаєте зробити?
Стерти диск і встановити Ubuntu
Увага: Це дія знищить усі Ваші програми, документи, фотографії, музику,та будь-які інші файли на усіх операційних системах.
Зашифрувати встановлену Ubuntu для підвищення захисту
Ключ захисту можна буде вибрати на наступному кроці.
Використовувати LVM для встановленої Ubuntu
За допомогою цього пункту можна наказати використовувати керування логічними томами (LVM). За допомогою такого керування можна буде створювати знятки системи та спростити зміну розмірів її розділів.
Щось інше
Ви можете створювати або міняти розміри розділів самотужки, або вибрати декілька розділів для Ubuntu.
Вийти
Назад
Встановити зараз

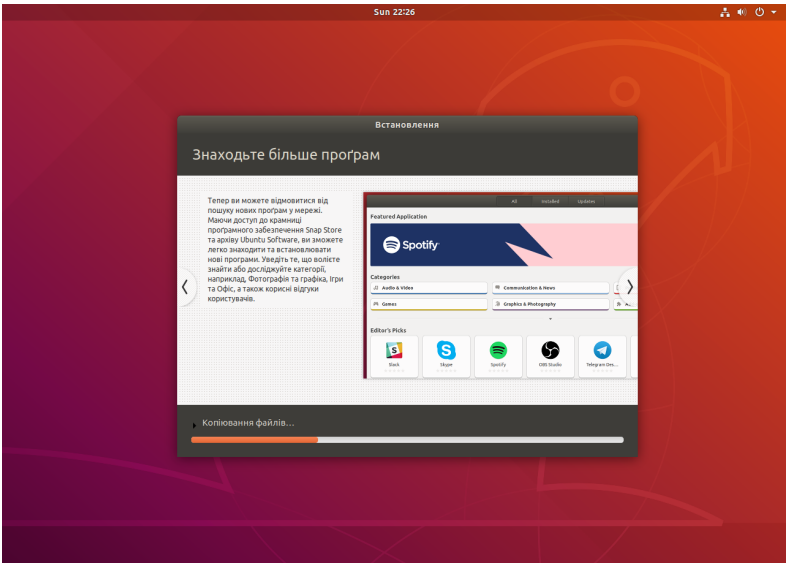

Sun 19:20
Встановлення
Тип встановлення
На Вашому комп'ютері не виявлено жодної операційної системи. Що Ви бажаєте зробити?
Стерти диск і встановити Ubuntu
Увага: Ця дія знищ
Зашифрувати в
Ключ захисту мож
Використовуват
За допомогою цьо
розділів.
Щось інше
Ви можете створю
Записати зміни на диски?
Якщо ви продовжите, то всі зміни, вказані нижче, будуть записані на диски. Інакше, ви зможете зробити додаткові зміни вручну.
Змінені таблиці розділів на наступних пристроях:
SCSI1 (0,0,0) (sda)
Наступні розділи будуть відформатовані:
розділ #1 пристрою SCSI1 (0,0,0) (sda) як ext4
Повернутися
Продовжити
ростити зміну розмірів її
Назад
Встановити зараз

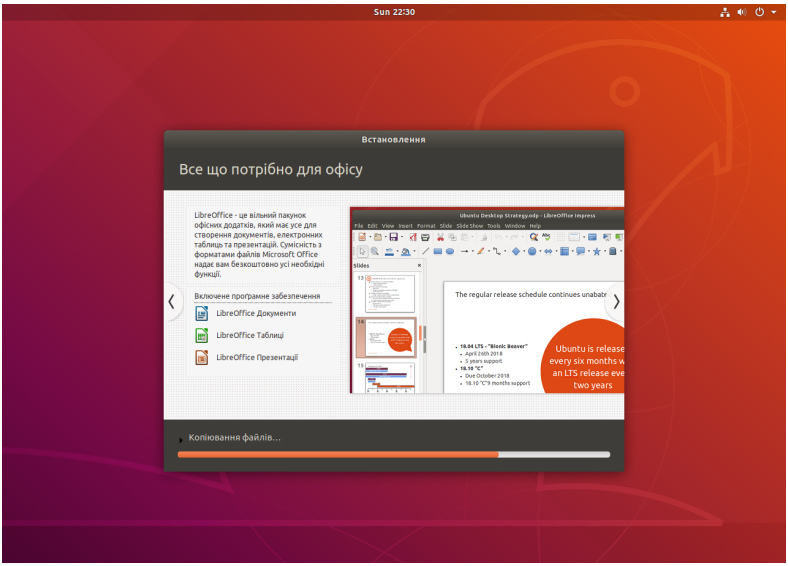

Sun 19:14
Встановлення
Ваше місцезнаходження?
Kiev
Назад
Продовжити


Sun 22:49
Встановлення
Хто ви?
Ваше ім'я:
user
Назва Вашого комп'ютера:
Ubuntu-PC
Назва, яка використовується для зв'язку з іншими комп'ютерами.
Увести ім'я користувача:
user
Оберіть пароль:
Підтвердження паролю:
Входити автоматично
Для входу потрібен пароль
Назад
Продовжити

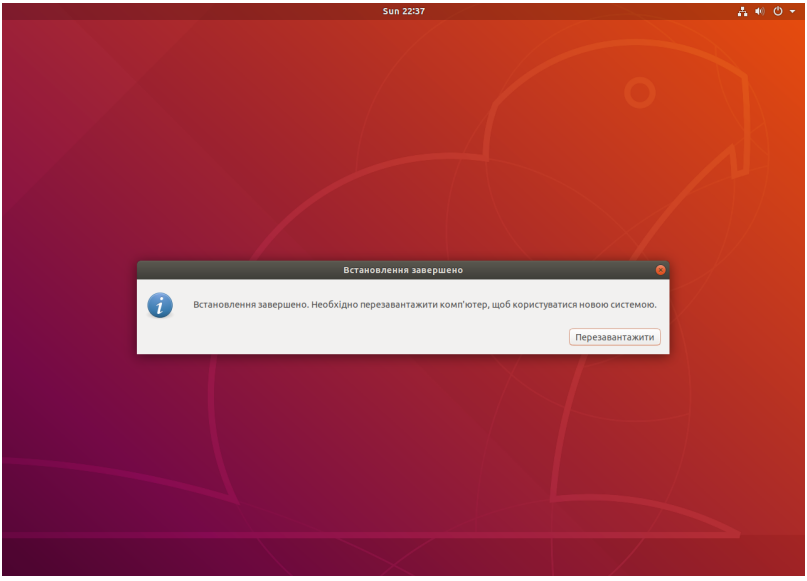

Sun 22:25
Встановлення
Копіювання файлів...
Skip

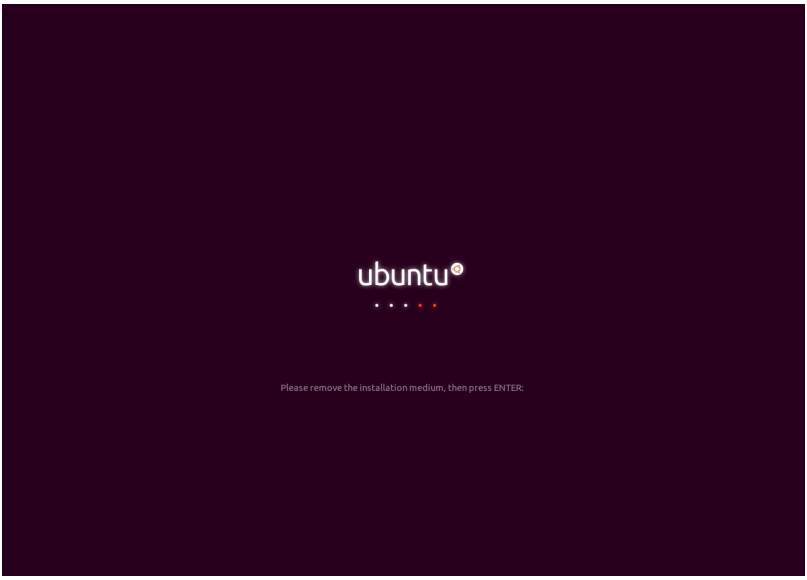
Далі треба дочекатися завершення встановлення.



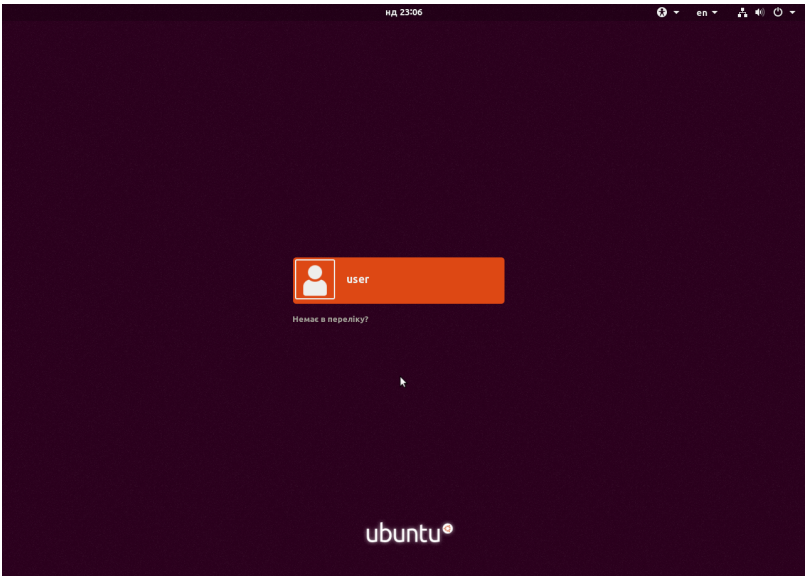



Sun 22:33
Встановлення
Довідка і підтримка
Офіційна документація покриває більшість з найпоширеніших питань щодо Ubuntu. Вона доступна як онлайн, так і через іконку Допомога в панелі запуску.
На сайті Ask Ubuntu ви можете задати питання або ознайомитися із широкою збіркою питань, на які вже дано відповіді. Підтримка вашою рідною мовою може надаватися відповідною командою місцевої спільноти.
Щоб отримати посилання на інші корисні ресурси, відвідайте Підтримку спільноти або Комерційну підтримку.
Налаштовування обладнання...

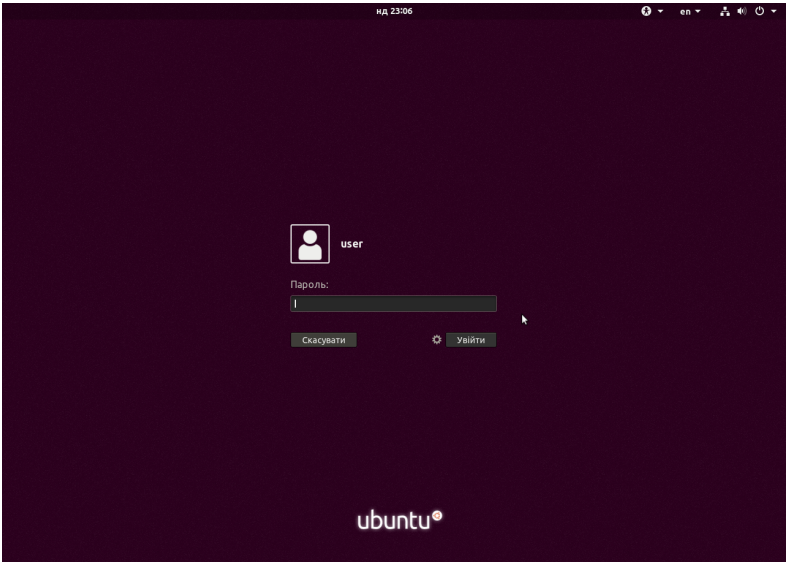

Sun 22:37
Встановлення завершено
Встановлення завершено. Необхідно перезавантажити комп'ютер, щоб користуватися новою системою.
Перезавантажити



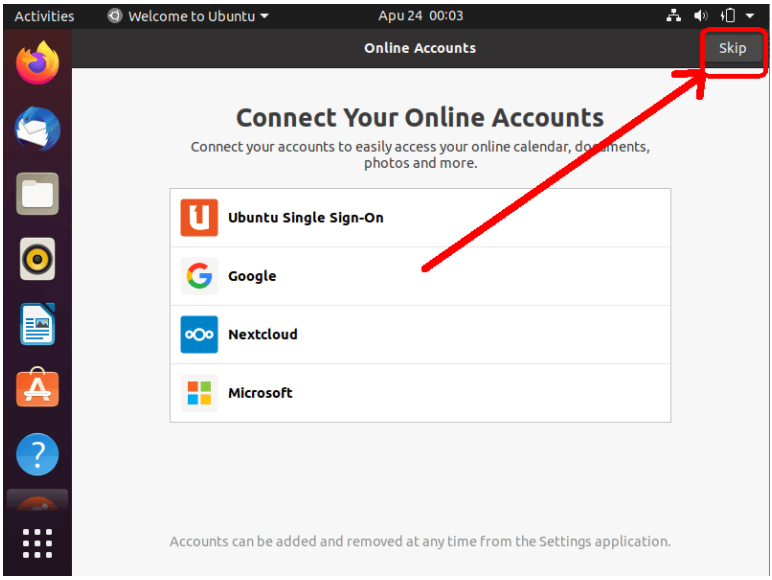
Коли встановлення завершиться, напискаємо Enter. Віртуальна машина почне свій перший запуск.

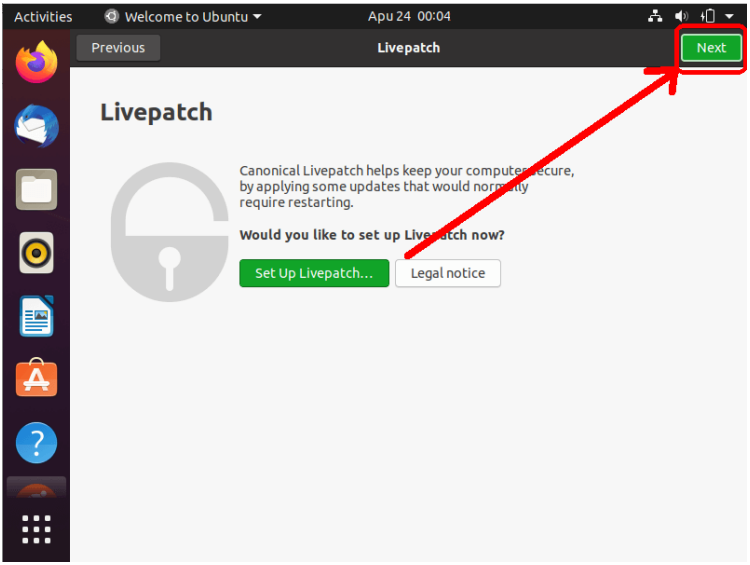


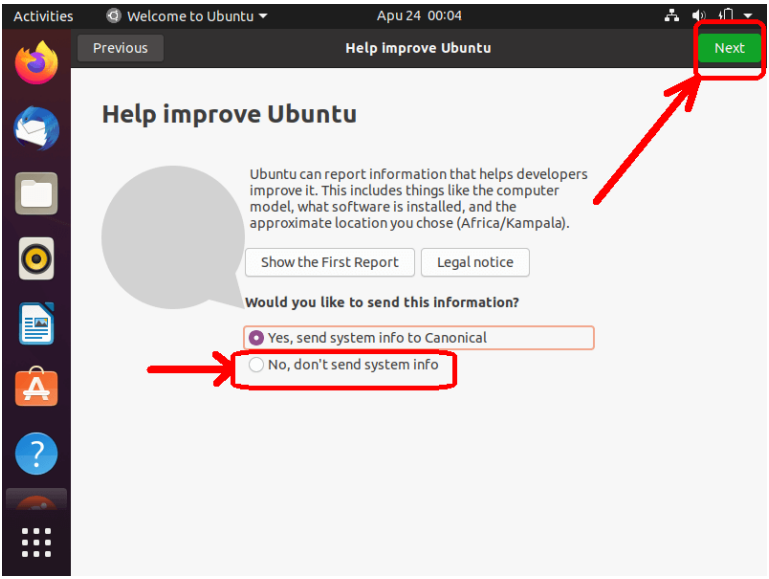
Вводимо пароль, який вказали під час встановлення Ubuntu.



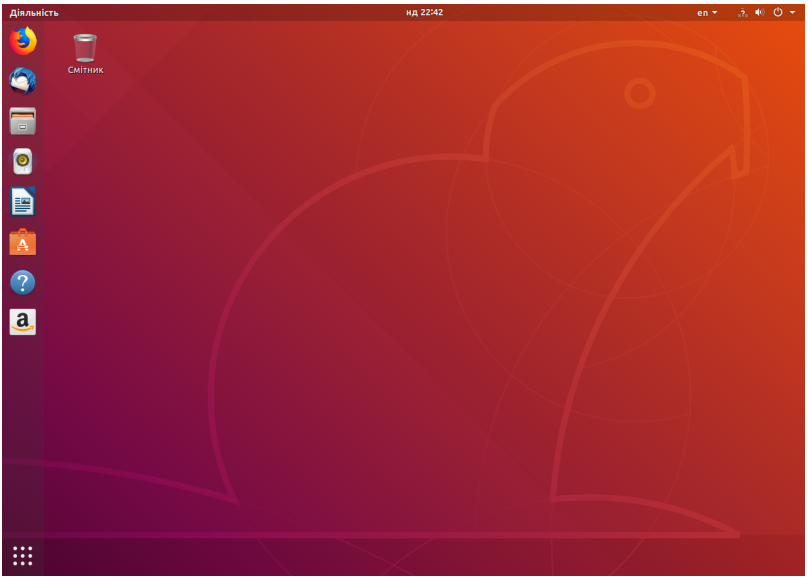
Додаткові налаштування можна пропустити.





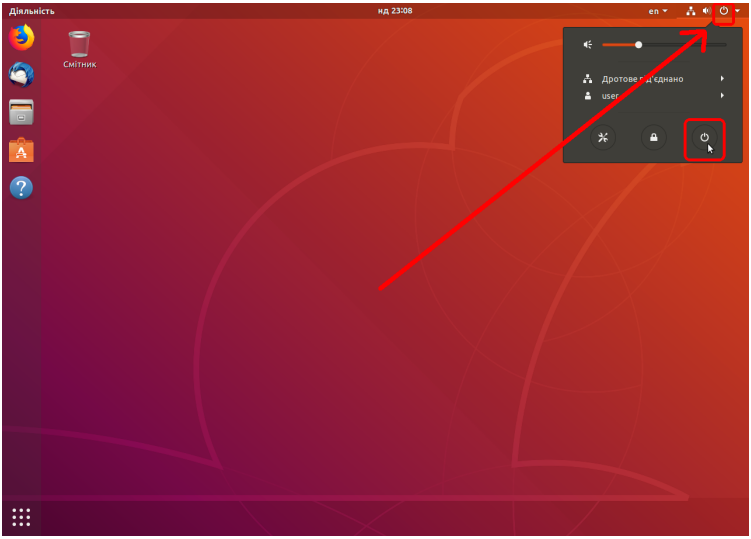


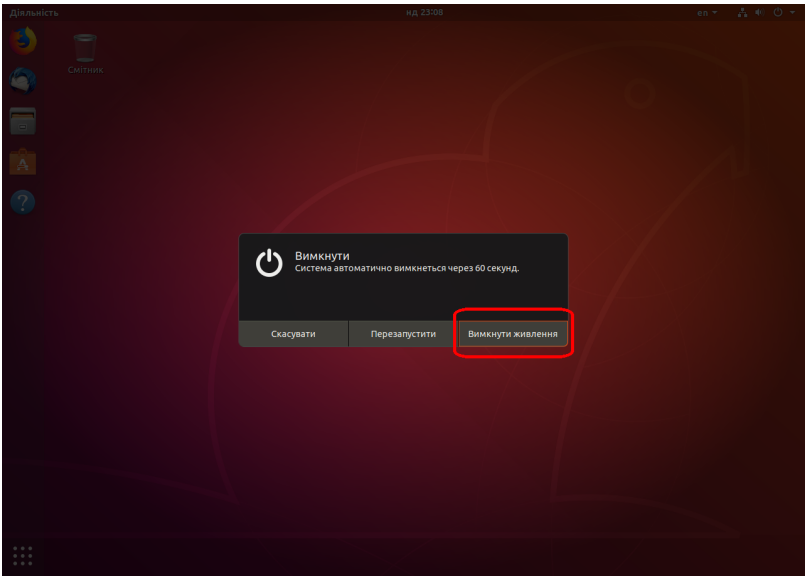
Ubuntu успішно запущено.



Меню додатків знаходиться у лівому нижньому кутку.

Як безпечно вимкнути віртуальну машину.





7.2 Лабораторна робота №2. Встановлення іншого дистрибутиву Linux паралельно з Ubuntu

Встановити інший дистрибутив Linux (згідно свого варіанту) на ту ж віртуальну машину, що й у попередній лабораторній роботі паралельно (пліч-о-пліч) із Ubuntu. Після цього при запуску віртуальної машини має з'явитися меню з вибором першої чи другої ОС.

Порада: якщо у 1-й лабораторній роботі ви створили віртуальну машину з жорстким диском менше 30 ГБ, треба його збільшити до 30, щоб вистачило місця для двох ОС.

Варіанти завдань:

1. Kubuntu

2. Lubuntu
3. Xubuntu
4. Ubuntu Kylin
5. Ubuntu Mate
6. Linux Mint Mate
7. Linux Mint Cinnamon
8. Linux Mint XFCE
9. KDE Neon
10. Fedora*
11. Debian*
12. Kali Linux*
13. CentOS*
14. Arch Linux**

7.3 Лабораторна робота №3. Запуск вебсервера Apache на Linux

Налаштувати й запустити веб-сервер Apache по черзі на обох операційних системах, встановлених у попередніх лабораторних роботах. Створити вебсторінку `index.html` (якщо її немає) і помістити її в директорію сервера Apache (в Ubuntu це `/var/www/html`). У цьому файлі написати своє прізвище та ім'я.

Щоб можна було доступитися до сервера віртуальної машини із зовнішнього комп'ютера (Windows), треба налаштувати для віртуальної машини NAT. Host port 8080, guest port 80, Host IP та Guest IP не вказувати. Це означає, що якщо у браузері Windows написати `localhost:8080`, VirtualBox перехопить сигнал і перенаправить його у віртуальну машину на порт 80, де його прийме сервер Apache.

7.3.1 Як встановити вебсервер Apache на Linux

Вебсервер — це програма, яка постійно запущена у фоновому режимі й "слухає"(перевіряє), чи хтось підключився до даного комп'ютера через мережу. Якщо так, то надсилає у відповідь "клієнтові"якийсь файл, наприклад, вебсторінку. Є різні безкоштовні вебсервери, наприклад, Apache, nginx, lighttpd. Ми будемо розглядати Apache, оскільки він найпростіший у налаштуванні.

Програми в Linux (також їх називають пакетами) легко встановлюються за допомогою командного рядка (консолі).

Спершу відкриємо консоль (термінал). В Ubuntu її можна відкрити комбінацією клавіш `Ctrl+Alt+T`. В інших системах Console (Terminal) можна знайти у меню програм.

У консолі спершу введемо команду для сканування усіх доступних репозиторіїв та їхніх пакетів. Додаємо префікс `sudo`, оскільки ця дія вимагає прав адміністратора.

На Debian, Ubuntu, Linux Mint:

```
sudo apt-get update
```

На Red Hat, Fedora, CentOS:

```
sudo yum update
```

Процес сканування може зайняти 1-2 хвилини. А далі встановлюємо пакет `apache2` (тобто сервер Apache версії 2) або `httpd`:

На Debian, Ubuntu, Linux Mint:

```
sudo apt-get install apache2
```

На Red Hat, Fedora, CentOS:

```
sudo yum install httpd
```

Якщо консоль запитає, чи дійсно встановити, натискаємо `Y` і потім `Enter`. Після встановлення сервер Apache на деяких системах може запускатися автоматично. Щоб перевірити, чи його запущено, вводимо команду:

На Debian, Ubuntu, Linux Mint:

```
sudo service apache2 status
```

На Red Hat, Fedora, CentOS:

```
sudo service httpd status
```

Якщо сервер не запущено, його можна запустити командою:

На Debian, Ubuntu, Linux Mint:

```
sudo service apache2 start
```

На Red Hat, Fedora, CentOS:

```
sudo service httpd start
```

Якщо потрібно зупинити його, замість `start` пишемо `stop`. Якщо потрібно перезапустити його (тобто зупинити і знову запустити), можна написати `restart`.

7.3.2 Як редагувати файл у консолі Linux

Коли ми відкриваємо консоль, то за замовчуванням знаходимось у директорії (папці) . У Linux символом "хвилька"позначають домашню директорію поточного користувача. Наприклад, якщо у користувача логін `user1`, то його домашня директорія буде `/home/user1`, або скорочено .

Щоб перейти до іншої директорії, треба ввести команду:

```
cd <шлях_до_іншої_директорії>
```

Це скорочення від англ. change directory (змінити директорію). Шлях може містити символ крапки (поточна директорія) або дві крапки підряд (батьківська директорія). Зверніть увагу, дві крапки (..) — це не двокрапка.

Приклад 1. Перейти до підпапки mydir, яка знаходиться в поточній папці:

```
cd mydir
```

або

```
cd ./mydir
```

Приклад 2. Перейти до батьківської директорії:

```
cd ..
```

Приклад 3. Перейти до домашньої директорії поточного користувача:

```
cd /home/<логін_користувача>
```

або

```
cd ~
```

Переглянути список файлів у поточній директорії можна командою:

```
ls
```

Але вона не виводить прихованих файлів. Щоб побачити приховані файли, треба писати:

```
ls -a
```

А щоб побачити розмір кожного файлу та коротку інформацію про нього:

```
ls -l
```

(від слова list — список). Ці дві опції можна комбінувати:

```
ls -al
```

або

```
ls -la
```

Для редагування файлів у консолі можна використовувати консольні текстові редактори `nano` або `vi`. Розглянемо `nano`, оскільки він простіший у користуванні.

Припустимо, треба відкрити для редагування файл `/var/www/html/index.html`. Для цього треба спочатку перейти до папки, де міститься файл `index.html`:

```
cd /var/www/html
```

А потім запустити текстовий редактор nano, вказавши, який файл редагувати. Оскільки файл знаходиться в системній папці `/var`, то для його редагування потрібні права адміністратора:

```
sudo nano index.html
```

Є й коротший спосіб. Можна не переходити до папки з потрібним файлом, а просто вказати редактору `nano`, де він знаходиться:

```
sudo nano /var/www/html/index.html
```

Файл відкриється для редагування. Стрілками на клавіатурі шукаємо потрібний рядок і редагуємо його. Ліва кнопка миші працювати в консольному редакторі не буде, але можна прокручувати файл колесом миші.

Щоб вийти, натискаємо `Ctrl+X`. Редактор запитає, чи зберегти файл. Натискаємо `Y` (або `N`, щоб не зберігати). Потім нас запитають, під якою назвою зберегти цей файл. Натискаємо `Enter`, щоб зберегти файл із такою ж назвою, яка була.

7.4 Лабораторна робота №4. Встановлення Ubuntu Server

1. Створити нову віртуальну машину.
2. Завантажити і встановити консольну версію (без графічного інтерфейсу) операційної системи Ubuntu – Ubuntu Server. Це файл у форматі ISO. При встановленні обов'язково вибрати англійську мову.
    - Якщо у вас 64-розрядний процесор, то завантажити найновішу версію Ubuntu Server можна з офіційного сайту.
    - Якщо ж 32-розрядний, то остання версія, яка підтримує їх, – Ubuntu Server 16.04, її можна також завантажити на офіційному сайті (вибрати Server install image -> 32-bit PC (i386) server install image).

3. Налаштувати і запустити сервер Apache на щойно встановленій системі у консольному режимі.

4. Знати основні команди Linux для роботи з файлами.

7.5 Лабораторна робота №5. Розгортання веб-сайту на віртуальному хості Apache через SSH

У попередній лабораторній роботі всі дії виконувались безпосередньо в Ubuntu Server на віртуальній машині. На практиці часто буває так, що сервер знаходиться далеко, а з ним працюють віддалено – через Інтернет за допомогою консолі. Ця технологія називається SSH.

У даній роботі введемо позначення: сервер – віртуальна машина з Ubuntu Server, створена у попередній лабораторній роботі, клієнт – ваша зовнішня операційна система, на якій працює комп'ютер (зазвичай це Windows).

1. Встановити на сервері програму для того, щоб можна було з ним працювати віддалено, тобто SSH-сервер, якщо його ще не встановлено (`sudo apt-get install openssh-server`). Перевірити, чи його запущено, можна командою `sudo service ssh status`.

2. Налаштувати NAT для SSH у віртуальній машині (порт 22). IP-адресу не вказувати. Будь-який SSH-сервер у Linux зазвичай працює на порті 22.

3. Налаштувати NAT для доступу до майбутнього сайту, наприклад, порт 8081, який перенаправляє на 81.

4. Завантажити на клієнті безкоштовний FTP-клієнт FileZilla (для перекидування файлів на сервер). А також, якщо у вас Windows, то також SSH-клієнт Putty (для керування сервером із консолі).

5. Підключитися до сервера з клієнта через Putty. Встановити на сервері PHP (`sudo apt-get install php libapache2-mod-php`). Подальші кроки на сервері робити через Putty.

6. На сервері налаштувати віртуальний хост на сервері Apache.

7. На клієнті завантажити одну систему керування вмістом (скорочено CMS), написану мовою PHP (згідно свого варіанту, наприклад, Joomla, Wordpress, phpBB, Drupal, OpenCart, MediaWiki).

8. За допомогою FileZilla перекинути архів із CMS на сервер. Розпакувати там за допомогою консолі (а не FileZilla) у директорію `/home/ваш_логін/www`.

9. Встановити на сервері MySQL-сервер (`sudo apt-get install mysql-server php-mysql`). Створити нового користувача MySQL і порожню базу даних.

10. Встановити на сервері phpMyAdmin. Після завершення встановлення ввести у браузері клієнта `localhost:8080/phpmyadmin`. У текстових полях ввести логін та пароль користувача MySQL, створеного на 9-му кроці. Після цього у списку має бути створена на 9-му кроці порожня база даних.

11. Запустити встановлення CMS у браузері на клієнті. При встановленні вказати хост `localhost`, ло-

гін та пароль створеного на 9-му кроці користувача MySQL і назву нової бази даних.

Лабораторна робота вважається виконаною, якщо CMS успішно працює в браузері клієнта.

7.5.1 Як розпакувати архів у Linux

ZIP Якщо архів у форматі `.zip`, то його можна розпакувати командою

```
unzip назва_архіву.zip
```

Тоді він розпакується в поточну директорію. Якщо потрібно вказати іншу директорію, тоді пишеться

```
unzip назва_архіву.zip -d шлях_до_директорії
```

Якщо виникає помилка, що команда `unzip` невідома, то треба встановити відповідний пакет:

```
sudo apt-get install unzip
```

Якщо ж виникає помилка, що пакет unzip неможливо знайти, слід підключити репозиторій (сервер, на якому знаходяться пакети) із назвою universe:

```
sudo add-apt-repository universe
```

TAR Якщо архів у форматі `.tar`, то його можна розпакувати командою

```
tar -xvf назва_архіву.tar
```

Тут опція `x` означає розпакувати (extract), `v` означає показувати детально (verbose), які файли розпаковуються, а після `f` пишеться назва архіву, який треба розпакувати.

TAR.GZ Задля економії дискового простору та Інтернет-трафіку файли зменшуються спеціальними програмами для стиснення даних, наприклад, `gzip` чи `bzip2`. Якщо архів у форматі `.tar.gz`, то його можна розпакувати командою

```
tar -xvzf назва_архіву.tar.gz
```

Тут опція `z` означає, що архів був стиснений програмою `gzip`.

TAR.BZ2 Якщо архів у форматі `.tar.bz2`, то його можна розпакувати командою

```
tar -xvjf назва_архіву.tar.bz2
```

Тут опція `j` означає, що архів був стиснений програмою `bzip2`.

7.5.2 Створення віртуального хоста Apache у Linux

Якщо на одному сервері Apache треба розмістити не один, а багато веб-сайтів, то використовують віруальні хости (virtual hosts).

Спершу дозволимо серверу Apache працювати на новому порті, наприклад, на 81-му (окрім стандартного 80-го). Для цього відредагуємо файл

```
sudo nano /etc/apache2/ports.conf
```

Після рядка

```
Listen 80
```

вставимо ще один

```
Listen 81
```

І зберігаємо.

Далі створимо директорію, де буде розміщено майбутній сайт. Зазвичай це роблять у домашній директорії користувача.

```
mkdir /home/ваш_логін/www
```

А також директорію для зберігання тимчасових файлів Apache і PHP, наприклад:

```
mkdir /home/ваш_логін/tmp
```

Встановимо права доступу 777 на ці папки.

```
chmod -R 777 /home/ваш_логін/www
chmod -R 777 /home/ваш_логін/tmp
```

Щоб створити новий віртуальний хост, треба перейти до наступної директорії (припускається, що Apache вже встановлено):

```
cd /etc/apache2/sites-available/
```

Тут для кожного віртуального хоста створюється окремий файл із розширенням `.conf`. Створимо новий:

```
sudo touch mysite.conf
```

Назва може бути довільною. Далі відредагуємо його

```
sudo nano mysite.conf
```

і заповнимо його наступним чином:

```
<VirtualHost *:80 *:81>
    ServerName mysite.com
    ServerAlias www.mysite.com
    ServerAdmin admin@mysite.com
    DocumentRoot /home/ваш_логін/www

    <Directory /home/ваш_логін/www>
        Options +FollowSymLinks +Indexes
        Require all granted
        AllowOverride All
    </Directory>

    <IfModule mpm_itk_module>
        AssignUserId ваш_логін ваш_логін
    </IfModule>
    php_admin_value open_basedir
     /home/ваш_логін/www:/home/ваш_логін/tmp
    php_admin_value upload_tmp_dir
     /home/ваш_логін/tmp
    php_admin_value session.save_path
     /home/ваш_логін/tmp

    ErrorLog /home/ваш_логін/tmp/error.log
    CustomLog /home/ваш_логін/tmp/access.log combined
</VirtualHost>
```

Не забудьте замінити `ваш_логін` на ім'я свого користувача Linux.

Після цього новий віртуальний хост активується командою

```
sudo a2ensite mysite.conf
```

Щоб зміни вступили в дію, слід повідомити Apache про зміни в його налаштуваннях:

```
sudo service apache2 reload
```

7.5.3 Встановлення phpMyAdmin в Ubuntu

Пакет `phpmyadmin` (зверніть увагу, що всі літери малі) встановлюється командою:

```
sudo apt-get install phpmyadmin
```

Якщо ж виникає помилка, що пакет `phpmyadmin` неможливо знайти, слід підключити репозиторій (сервер, на якому знаходяться пакети) із назвою `universe`:

```
sudo add-apt-repository universe
```

При встановленні `phpmyadmin` треба вибрати зі списку сервер `apache2`, натиснувши пробіл, а потім OK.

Якщо з'явиться вікно із запитанням, чи налаштувати допоміжні таблиці, можна обрати No.

Після встановлення `phpmyadmin` слід відкрити у браузері сторінку `localhost:8080/phpmyadmin`.

Якщо виникає помилка "Not Found це означає, що сервер Apache не бачить конфігураційного файлу phpMyAdmin. Тоді треба в кінці файлу `/etc/apache2/apache2.conf` додати наступний рядок

```
Include /etc/phpmyadmin/apache.conf
```

і перезапустити сервер Apache командою

```
sudo service apache2 restart
```

7.5.4 Створення нового користувача MySQL в Ubuntu

Спочатку в консолі підключаємось до MySQL командою

```
sudo mysql -u root
```

Консоль перейде в режим введення команд для MySQL. Тоді створюємо нового користувача для MySQL.

```
CREATE USER 'new_user'@'localhost'
  IDENTIFIED BY 'new_password';
```

Зверніть увагу: в кінці команд MySQL крапка з комою обов'язкова. Замість `new_user` підставляємо логін нового користувача, а замість `new_password` – його пароль. Ці логін і пароль потім знадобляться при встановленні сайту.

Далі створюємо нову базу даних.

```
CREATE DATABASE mysite CHARACTER SET utf8
  COLLATE utf8_unicode_ci;
```

Замість `mysite` можна підставити будь-яку назву для бази даних.

Переконаємося, що базу успішно створено. Введемо

```
SHOW DATABASES;
```

БД mysite має бути у списку.

Після цього надамо повний доступ до цієї БД для новоствореного користувача.

```
GRANT ALL ON mysite.* TO 'new_user'@'localhost';
```

Замість `new_user` підставляємо логін створеного вище користувача.

Щоб зміни вступили в дію, треба ввести

```
FLUSH PRIVILEGES;
```

Виходимо з режиму MySQL командою

```
exit;
```

7.6 Лабораторна робота №6. Знайомство з безкоштовними CMS

1. Розібратися з основними можливостями встановленої у попередній лабораторній роботі CMS (наприклад, створення та редагування публікацій, їхніх категорій).

2. Завантажити і встановити новий безкоштовний шаблон (тему, дизайн) для даної CMS.

3. Вивчити основні команди Linux для створення та редагування користувачів і груп.

Примітка: якщо при встановленні шаблону чи теми виникає помилка, що не вказано тимчасової директорії, слід встановити права доступу 777 на папку `tmp`:

```
sudo chmod -R 777 /home/ваш_логін/tmp
```

7.7 Лабораторна робота №7. Встановлення Java-сервера Tomcat

Встановити на Ubuntu Java-сервер Apache Tomcat (не плутати зі звичайним сервером Apache для PHP).

Примітка: Tomcat працюватиме на порті 8080 на сервері (у віртуальній машині). Тож треба налаштувати на клієнті NAT. Наприклад, щоб порт 8082 (чи якийсь інший) переадресовував на порт 8080 у віртуальній машині.

7.7.1 Як встановити Apache Tomcat в Ubuntu

Apache Tomcat — це вебсервер і контейнер сервлетів, який дозволяє запускати Java вебдодатки. У цьому посібнику описано встановлення та базове налаштування Tomcat на Ubuntu Server.

Крок 1: Оновлення системи Перед початком оновимо список пакетів і саму систему:

```
sudo apt update
sudo apt upgrade
```

Крок 2: Встановлення Java Tomcat вимагає встановленого середовища Java. Встановимо OpenJDK:

```
sudo apt install default-jdk
```

Перевіримо версію Java:

```
java -version
```

Крок 3: Завантаження Tomcat Перейдемо на офіційний сайт Tomcat: `https://tomcat.apache.org`

Завантажимо останню стабільну версію (наприклад, Tomcat 10.x):

```
cd /tmp
wget https://downloads.apache.org/tomcat/tomcat-10/
  v10.1.20/bin/apache-tomcat-10.1.20.tar.gz
```

Розпакуємо архів і перемістимо в стандартну директорію:

```
sudo tar -xzf apache-tomcat-10.1.20.tar.gz -C /opt/
sudo mv /opt/apache-tomcat-10.1.20 /opt/tomcat
```

Крок 4: Створення користувача Tomcat З міркувань безпеки створимо окремого користувача:

```
sudo useradd -m -U -d /opt/tomcat -s /bin/false tomcat
sudo chown -R tomcat: /opt/tomcat
```

Крок 5: Створення systemd-сервісу Створимо юніт-файл для systemd:

```
sudo nano /etc/systemd/system/tomcat.service
```

Вставимо наступне:

```
[Unit]
Description=Apache Tomcat Web Application Container
After=network.target

[Service]
Type=forking

User=tomcat
Group=tomcat

Environment="JAVA_HOME=
  /usr/lib/jvm/java-11-openjdk-amd64"
Environment="CATALINA_PID=
  /opt/tomcat/temp/tomcat.pid"
Environment="CATALINA_HOME=/opt/tomcat"
Environment="CATALINA_BASE=/opt/tomcat"
Environment="CATALINA_OPTS=
  -Xms512M -Xmx1024M -server -XX:+UseParallelGC"
Environment="JAVA_OPTS=-Djava.awt.headless=true
  -Djava.security.egd=file:/dev/./urandom"

ExecStart=/opt/tomcat/bin/startup.sh
```

```
ExecStop=/opt/tomcat/bin/shutdown.sh

[Install]
WantedBy=multi-user.target
```

Збережемо файл і активуємо сервіс:

```
sudo systemctl daemon-reexec
sudo systemctl daemon-reload
sudo systemctl enable tomcat
sudo systemctl start tomcat
```

Крок 6: Доступ до Tomcat через браузер Відкрийте браузер і перейдіть за адресою:

```
http://localhost:8080
```

Якщо все зроблено правильно — з'явиться домашня сторінка Apache Tomcat.

Крок 7: Увімкнення адмін-панелі Відредагуємо файл:

```
/opt/tomcat/conf/tomcat-users.xml
```

Додамо всередині тега `<tomcat-users>`:

```
<role rolename="manager-gui"/>
<role rolename="admin-gui"/>
<user username="admin" password="admin_password"
  roles="manager-gui,admin-gui"/>
```

Після цього перезапустимо Tomcat:

```
sudo systemctl restart tomcat
```

Тепер сервер Apache Tomcat встановлено, налаштовано і він готовий до розгортання Java-додатків.

7.8 Лабораторна робота №8. Встановлення .NET

1. Встановити .NET (або .NET Core) на Ubuntu Server.

2. Скомпілювати просту консольну програму мовою C# і запустити її.

7.8.1 Як встановити .NET в Ubuntu

.NET SDK дозволяє розробляти та запускати .NET-додатки на Linux.

Перед встановленням оновимо пакети системи:

```
sudo apt update
sudo apt upgrade
```

Додамо офіційний ключ Microsoft і репозиторій для Ubuntu:

```
wget https://packages.microsoft.com/config/ubuntu/
  $(lsb_release -rs)/packages-microsoft-prod.deb
sudo dpkg -i packages-microsoft-prod.deb
sudo apt update
```

Тепер встановимо .NET SDK (наприклад, версії 8):

```
sudo apt install dotnet-sdk-8.0
```

Перевіримо, що .NET встановлено коректно:

```
dotnet --version
```

Створимо консольний додаток для перевірки:

```
mkdir myapp
cd myapp
dotnet new console
dotnet run
```

Тепер маємо встановлене та готове до роботи середовище .NET SDK. Можна створювати, компілювати та запускати .NET-додатки безпосередньо на сервері Ubuntu.