Міністерство освіти і науки України
Український державний університет науки і технологій

Факультет «Комп'ютерні технології і системи»
(назва факультету)

Кафедра «Електронні обчислювальні машини»
(повна назва кафедри)

Пояснювальна записка
до кваліфікаційної роботи
бакалавра
(ступінь вищої освіти)

на тему: Вдосконалення механізмів інформаційної безпеки комп'ютерної мережі кафедри ЕОМ. Вдосконалення програмного забезпечення мережі
за освітньою програмою Кібербезпека
зі спеціальності: 125 Кібербезпека
(шифр і назва спеціальності)

Виконав: студент групи: КБ20120

(підпис студента) / Богдан КУШНІР /
(Ім'я ПРІЗВИЩЕ)

Керівник: (підпис) / Ігор ЖУКОВИЦЬКИЙ /
(посада, Ім'я ПРІЗВИЩЕ)

Нормоконтролер: (підпис) / Володимир ШАПОВАЛОВ /
(посада, Ім'я ПРІЗВИЩЕ)

Консультанти:

(назва розділу) (підпис) / / (посада, Ім'я ПРІЗВИЩЕ)

(назва розділу) (підпис) / / (посада, Ім'я ПРІЗВИЩЕ)

(назва розділу) (підпис) / / (посада, Ім'я ПРІЗВИЩЕ)

(назва розділу) (підпис) / / (посада, Ім'я ПРІЗВИЩЕ)

Засвідчую, що у цій роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент (підпис)

Дніпро – 2023 рік

Ministry of Education and Science of Ukraine
Ukrainian State University of Science and Technologies

Faculty of Computer Technologies and Systems
(faculty)

Department of Electronic Computing Machines
(department)

Explanatory Note
to Master's Thesis
bachelor
(higher education degree)

on the topic: Improvement of information security mechanisms of the computer network of the Department of Electronic Computing Machines. Improvement of the software in network.

according to educational curriculum_Cybersecurity

in the Speciality: 125 Cybersecurity
(speciality and its code)

Done by the student of the group: KB20120 / Bohdan KUSHNIR /
(name, surname)

Scientific Supervisor: / Ihor ZHUKOVYTSKYI /

(position, name, surname)

Normative controller : / Volodymyr SHAPOVALOV /
(position, name, surname)

Supervisors

(Chapter title heading)	/ / (position, name, surname)
(Chapter title heading)	/ / (position, name, surname)
(Chapter title heading)	/ / (position, name, surname)
(Chapter title heading)	/ / (position, name, surname)

Dnipro – 2023

Міністерство освіти і науки України
Український державний університет науки і технологій

Факультет: Комп’ютерні технології і системи
Кафедра: Електронні обчислювальні системи
Рівень вищої освіти: Перший (бакалаврський)
Освітня програма: Комп’ютерна інженерія
Спеціальність: 125 Кібербезпека
(шифр та назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри ЕОМ

_____ Ігор ЖУКОВИЦЬКИЙ
(підпис) (Ім’я ПРІЗВИЩЕ)

Дата _____

ЗАВДАННЯ

на кваліфікаційну роботу бакалавра
(ступінь вищої освіти)

студенту Кушніру Богдану Тарасовичу
(Прізвище, Ім’я По батькові)

1. Тема роботи: "Вдосконалення механізмів інформаційної безпеки комп'ютерної мережі кафедри ЕОМ. Вдосконалення програмного забезпечення мережі

Керівник роботи: Жуковицький Ігор Володимирович
(Прізвище, Ім’я, По батькові, науковий ступінь, вчене звання)

затверджені наказом від "01" 03 2023 р. № 198ст

2. Строк подання студентом роботи: 20.06.2023 р.

3. Вихідні дані до роботи: Існуюча структура мережі кафедри ЕОМ

4. Зміст пояснювальної записки (перелік питань, які потрібно опрацювати):

4.1. Існуюча структура та ПО мережі. Пропозиції щодо ii вдосконалення.

4.2. Встановлення та налаштування операційної системи.

4.3. Налаштування сервисів і безпеки

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Вступ	17.04.2023	5%
2	Існуюча структура та ПО мережі. Пропозиції щодо її вдосконалення.	27.04.2023	10%
3	Встановлення та налаштування операційної системи	12.05.2023	35%
4	Налаштування сервісів і безпеки	22.05.2023	35%
5	Висновки	16.06.2023	5%
6	Подання кваліфікаційної роботи до кафедри	19.06.2023	
7	Захист кваліфікаційної роботи на засіданні Екзаменаційної комісії	28.06.2023	

Студент _____ Богдан КУШНІР
(підпис) (Ім'я ПРІЗВИЩЕ)

Керівник роботи _____ Ігор ЖУКОВИЦЬКИЙ
(підпис) (Ім'я ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи бакалавра: 130 с., 3 рис., 2 табл., 25 додатків, 61 джерело.

Об’єкт розробки — програмне забезпечення кафедрального сервера.

Мета роботи — запровадити в мережі кафедри мінімальний набір функцій та сервісів характерних сучасній мережі.

Проведено огляд наявної архітектури мережі кафедри. Її апаратного та програмного забезпечення. Прийнято рішення про модернізацію всього програмного забезпечення сервера. Здійснено огляд сімейств операційних систем Windows і Linux. Проведено огляд різноманітних дистрибутивів з сімейства Linux. Виконано встановлення дистрибутиву Manjaro на сервер у мінімально необхідній конфігурації. Виконано встановлення та налаштування базових мережевих сервісів:

— Dynamic Host Configuration Protocol (DHCP);
— Domain Name System (DNS);
— Network Time Protocol (NTP).

Наведено відповідні коментарі та конфігурації. Виконано синхронізацію DHCP і DNS за допомогою технології Dynamic Domain Name System (DDNS). Налаштовано прості й ефективні правила фільтрації мережевого трафіку. Налаштовано систему контейнеризації Linux Containers (LXC), для надання можливості подальшого масштабування сервісів сервера.

Розроблена конфігурація сервера є простою та надійною. Вона забезпечує високу гнучкість та надійність роботи мережі. Розроблена мережа є досить гнучкою та адаптивною.

Ключові слова: СЕРВЕР, LINUX, MANJARO, DHCP, DNS, DDNS, SECURE SHELL (SSH), КОНТЕЙНЕРИЗАЦІЯ, LXC, NETFILTER, NFTABLES, FIREWALL, БРАНДМАУЕР, МЕРЕЖА.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

AUR Arch User Repository. Це сховище, кероване спільнотою для користувачів Arch. Він містить описи пакунків (PKGBUILDs), які дозволяють скомпілювати пакет із вихідного коду за допомогою makepkg, а потім встановити його за допомогою pacman. AUR було створено, щоб упорядковувати та ділитися новими пакетами зі спільноти, а також допомагати пришвидшувати включення популярних пакетів у додатковий репозиторій. У цьому документі пояснюється, як користувачі можуть отримати доступ і використовувати AUR.

BIND Berkeley Internet Name Domain (раніше Berkeley Internet Name Daemon). Універсальний каркас для побудови елементів інтернет-інфраструктури, таких як сервери DNS і DHCP; відкрита і найпоширеніша реалізація DNS-сервера, що забезпечує виконання перетворення DNS-імені в IP-адресу і навпаки. BIND підтримується організацією Internet Systems Consortium. 10 з 13 кореневих серверів DNS працюють на BIND, решта 3 працюють на NSD.

BIOS Basic Input/Output System. Тип вбудованої програми, що зберігається у постійній пам'яті і виконує початкову ініціалізацію машини після її увімкнення, а також надає спеціальні точки входу для сервісних процедур, що можуть використовуватися операційною системою. Фізично код BIOS записаний у мікросхемах постійної або флеш-пам'яті, розташованих на системній платі комп'ютера. Назва походить від частини операційної системи CP/M. Перші BIOS для комп'ютерів IBM PC були пропрієтарним програмним забезпеченням, однак стороннім компаніям вдалося розробити власні версії, частково шляхом зворотної розробки. Інтерфейс програм оригінальної BIOS довгий час залишався стандартом де-факто.

BSD Berkeley Software Distribution. Назва кількох POSIX-сумісних операційних систем сімейства UNIX, створених на основі розробок Каліфорнійського університету в Берклі на початку 1970-х років. Безкоштовні, мають відкритий початковий код. Сьогодні найпопулярнішою є FreeBSD. Назва *BSD також є збірною для сучасних наступників тих збірок.

DDNS Dynamic Domain Name System. Метод автоматичного оновлення сервера імен у системі доменних імен, часто в реальному часі, з активною конфігурацією DDNS налаштованих імен хостів, адрес або іншої інформації.

DHCP Dynamic Host Configuration Protocols. Стандартний протокол прикладного

рівня, який дозволяє комп'ютерам автоматично отримувати IP-адресу та інші параметри, необхідні для роботи в мережі.

DNS Domain Name System. Ієрархічна розподілена система перетворення імені хоста (комп'ютера або іншого мережевого пристрою) в IP-адресу.

DOS Disk Operating System. Операційна система, яка розміщується на дисковому пристрої зберігання даних, такому як дискета, жорсткий диск або оптичний диск, і може використовувати його. Дискова операційна система забезпечує файлову систему для організації, читання та запису файлів на диску зберігання. Строго кажучи, це визначення не включає жодних інших функціональних можливостей, тому воно не застосовується до складніших операційних систем, таких як Microsoft Windows, і його більш доречно використовувати лише для старих поколінь операційних систем. Дискові операційні системи для мейнфреймів, міні-комп'ютерів, мікропроцесорів і домашніх комп'ютерів зазвичай завантажуються з дисків як частина процесу завантаження.

FHS Filesystem Hierarchy Standard. Стандарт прийнятий для уніфікації розташування файлів і каталогів загального призначення у файловій системі операційної системи UNIX. Сьогодні більшість UNIX-подібних систем в тій або іншій мірі слідують цим правилам. Наприклад, типова база даних про користувачів завжди зберігається у файлі /etc/passwd. Поточна версія стандарту — 3., анонсована 3 червня 2015 р.

GPS Global Positioning System. Сукупність радіоелектронних засобів, що дозволяє визначати положення та швидкість руху об'єкта на поверхні Землі або в атмосфері. Положення об'єкта обчислюється завдяки використанню розміщеного на ньому GPS-приймача, який приймає та обробляє сигнали супутників космічного сегменту GPS-системи глобального позиціонування. Для визначення точних параметрів орбіт супутників та керування GPS-системою вона в своєму складі має наземні центри управління.

GPT GUID Partition Table. Стандарт формату розміщення таблиць розділів на фізичному жорсткому диску. Він є частиною UEFI. EFI використовує GPT там, де BIOS використовує MBR. На відміну від MBR, яка починається з машинного коду, що шукає і завантажує активний розділ, GPT використовує розширені можливості EFI для здійснення цих процесів. Проте MBR присутня в самому початку диску (блок LBA 0) як для захисту, так і з метою сумісно-

сті. Саме GPT починається з Partition Table Header.

GRUB Grand Unified Bootloader 2. Завантажувач операційної системи від проекту GNU. GRUB дозволяє користувачеві мати кілька встановлених операційних систем і при вмиканні комп'ютера вибирати одну з них для завантаження. Перша версія GRUB наразі не використовується і носить іншу назву — GRUB Legacy.

IoT Internet of Things. Концепція мережі, що складається із взаємозв'язаних фізичних пристроїв, які мають вбудовані давачі, а також програмне забезпечення, що дозволяє здійснювати передачу і обмін даними між фізичним світом і комп'ютерними системами в автоматичному режимі, за допомогою використання стандартних протоколів зв'язку. Окрім давачів, мережа може мати виконавчі пристрої, вбудовані у фізичні об'єкти і пов'язані між собою через дротові чи бездротові мережі. Ці взаємопов'язані пристрої мають можливість зчитування та приведення в дію, функцію програмування та ідентифікації, а також дозволяють виключити необхідність участі людини, за рахунок використання інтелектуальних інтерфейсів.

IPv4 Internet Protocol version 4. Четверта версія мережевого протоколу IP. Перша версія протоколу, яка набула широко розповсюдження. Протокол IPv4, описаний у RFC 791 (вересень 1981 року), прийшов на заміну описаному у RFC 760 (січень 1980 року). Використовує 4 байтну форму запису адрес пристроїв в комп'ютерній мережі.

IPv6 Internet Protocol version 6. IP версії 6. Розробка протоколу IPv6 почалася 1992 року, а з 2003 р. його підтримку забезпечують виробники більшості телекомунікаційного устаткування (корпоративного рівня). IPv6 — новий крок у розвитку Інтернету. Цей протокол розроблено з урахуванням вимог до Глобальної мережі, що постійно зростають. 3 лютого 2011 року IANA виділила останні п'ять блоків IP-адрес /8 (IPv4). Найбільш суттєва різниця між IPv4 та IPv6 полягає в тому, що раніше на інтернет-адресу виділяли 4 байти (32 біти), що відповідає стандартній на сьогодні чотириблоковій адресі IP, а протокол IPv6 виділяє на адресу 16 байтів (128 бітів). Це відповідає 340 секстильйонам адрес (3,4x1038) або по 5x1028 адрес на кожну людину.

ISC Internet Systems Consortium. Некомерційна організація, що займається підтримкою інфраструктури всесвітньої глобальної мережі інтернет. ISC займається розробкою і підтримкою відкритого програмного забезпечення.

JSON JavaScript Object Notation. Текстовий формат обміну даними між комп'ютерами. JSON базується на тексті, може бути прочитаним людиною. Формат дає змогу описувати об'єкти та інші структури даних. Цей формат використовується переважно для передавання структурованої інформації через мережу (завдяки процесу, що називають серіалізацією).

LSB Linux Standard Base. Спільний проєкт кількох дистрибутивів GNU/Linux з організацією Linux Foundation, метою якого є стандартизація внутрішньої структури операційних систем, заснованих на Linux. LSB спирається на існуючі специфікації, такі як POSIX, Single UNIX Specification, та інші відкриті стандарти, розширюючи і доповнюючи їх.

LXC Linux Containers. Система віртуалізації на рівні операційної системи для запуску декількох ізольованих примірників операційної системи Linux на одному комп'ютері. LXC не використовує віртуальні машини, а створює віртуальне оточення з власним простором процесів і мережевим стеком. Усі примірники LXC використовують один примірник ядра.

MBR Master Boot Record. Це перший фізичний сектор на жорсткому диску або іншому носії інформації, якщо цей пристрій розподіляється на логічні диски (розділи). MBR містить таблицю розділів (partition table) і невеликий фрагмент виконуваного коду. Мета MBR — не завантаження ОС, а всього лише вибір, з якого розділу саме відбувається завантаження.

NCR Name Change Request. DNS запит від клієнта до сервера для реєстрації або заміни доменного імені в зоні. Зазвичай всі подібні запити підписуються за допомогою TSIG. Це необхідно для автентифікації клієнта та перевірки автентичності запиту.

NTP Network Time Protocol. Мережевий протокол синхронізації внутрішнього годинника комп'ютера з використанням мереж зі змінною латентністю, заснований на комутації пакетів.

ODF Open Document Format. Відкритий формат файлів документів для зберігання й обміну офісними документами, доступними для редагування, в тому числі текстовими документами (такими як нотатки, звіти й книги), електронними таблицями, малюнками, базами даних, презентаціями. ODF являє собою не тільки файловий формат для зберігання документів, заснований на XML і незалежний від програм і платформ, але і набір вимог

до організації читання, запису і обробки подібних документів у застосунках. Цей стандарт розроблений індустріальною спільнотою OASIS, що займається розробкою і просуванням відкритих стандартів, і базується на XML-форматі, первісно створеному для OpenOffice.org. 3 травня 2006 року прийнятий як міжнародний стандарт ISO/IEC 26300.

POSIX Portable Operating System Interface. Набір стандартів, які описують інтерфейси між операційною системою та застосунками. Стандарт створений для забезпечення сумісності різних UNIX-подібних операційних систем та переносимості прикладних програм на рівні початкового коду програм.

RAID Redundant Array of Independent Disks. Технологія віртуалізації даних, яка об'єднує кілька дисків в логічний елемент для надійності збереження інформації та підвищення продуктивності накопичувачів. Дисковий масив — це набір дискових пристроїв, які працюють разом, щоб підвищити швидкість і/або надійність системи вводу/виводу. Цим набором пристроїв керує особливий RAID-контролер (контролер масиву), який забезпечує функції розміщення даних масивом; а для решти усієї системи дозволяє представляти увесь масив, як один логічний пристрій вводу/виводу. Завдяки паралельному виконанню операцій читання та запису на кількох дисках, масив забезпечує підвищену швидкість обміну у порівнянні з одним великим диском. Масиви також можуть забезпечувати надійне зберігання даних, для того, щоб дані не були втрачені у разі виходу з ладу одного з дисків. Залежно від рівня RAID, проводиться або дублювання або рівномірний розподіл даних на дисках.

RHEL RedHat Enterprise Linux. Дистрибутив Linux виробництва компанії Red Hat, орієнтований на комерційний ринок, включно з мейнфреймами. Red Hat здійснює підтримку кожної версії RHEL протягом 7 років з моменту її виходу. Уся офіційна підтримка Red Hat, і всі курси навчання та видача сертифікатів для розгортання апаратного та програмного забезпечення — Red Hat Certified Technician (RHCT), Red Hat Certified Engineer (RHCE), Red Hat Certified Security Specialist (RHCSS) та Red Hat Certified Architect (RHCA) складають основу платформи Red Hat Enterprise Linux.

SATA Serial AT Attachment. Послідовний інтерфейс обміну даними з накопичувачами інформації (як правило, з жорсткими дисками). SATA є розвитком

інтерфейсу ATA (IDE), який після появи SATA був перейменований в PATA (Parallel ATA).

SCSI Small Computer Systems Interface. Інтерфейс, розроблений для об'єднання на одній шині різних за своїм призначенням пристроїв, таких як тверді диски, накопичувачі на магнітооптичних дисках, приводи CD, DVD, стримери, сканери, принтери тощо. Раніше мав неофіційну назву Shugart Computer Systems Interface на честь творця Алана Ф. Шугарта. Теоретично можливий випуск пристрою будь-якого типу на шині SCSI. Після стандартизації в 1986 році, SCSI почав широко застосовуватися в комп'ютерах Apple Macintosh, Sun Microsystems. У персональних комп'ютерах, сумісних з IBM PC, SCSI не користується такою популярністю у зв'язку з своєю складністю і порівняно високою вартістю, і застосовується переважно в серверах.

SSD Solid-State Drive. Комп'ютерний запам'ятовувальний пристрій на основі мікросхем пам'яті та контролера керування ними, що не містить рухомих механічних частин.

SSH Secure Shell. Мережевий протокол рівня застосунків, що дозволяє проводити віддалене управління комп'ютером і тунелювання TCP-з'єднань (наприклад, для передачі файлів). Схожий за функціональністю з протоколом Telnet і rlogin, проте шифрує весь трафік, в тому числі і паролі, що передаються.

SSO Virtual Private Network. Технологія, що дозволяє користувачеві переходити з одного онлайн-сервісу до іншого, без повторної автентифікації, за єдиним ID. Наприклад, у великих приватних мережах часто існує декілька незалежних підсистем. З допомогою SSO реалізується доступ до всіх підсистем без повторного введення логіну/паролю. Для цього користувачеві буде достатньо ввести логін/пароль лише раз для однієї з підсистем і він матиме доступ до всіх інших.

SUS Single UNIX Specification. Загальна назва для сімейства стандартів, які повинна задовольняти операційна система, щоб називатися "Unix". SUS розробляється і підтримується Austin Group на основі попередніх розробок IEEE і The Open Group.

SWAP Один з механізмів віртуальної пам'яті, при якому окремі фрагменти оперативної пам'яті (зазвичай неактивні) переміщуються на жорсткий диск, звільняючи місце у оперативній пам'яті для завантаження інших фрагментів. Такими фрагментами в сучасних комп'ютерах є сторінки пам'яті. Тимчасово

вивантажені з пам'яті сторінки можуть зберігатися на зовнішніх носіях інформації як у файлі, так і в спеціальному розділі на жорсткому диску (англ. partition), звані відповідно swap-файл і swap-розділ. У разі відкачування сторінок, що відповідають вмісту будь-якого файлу (наприклад, англ. memory-mapped files), вони можуть вилучатися. При запиті такої сторінки вона може бути прочитана з оригінального файлу. Коли застосунок звернеться до відкачаної сторінки, відбудеться виняткова ситуація Page Fault. Обробник цієї події повинен перевірити, чи було раніше відновлено запитану сторінку, і, якщо вона є в swap, завантажити її назад в пам'ять.

UEFI Unified Extensible Firmware Interface. Стара назва — Extensible Firmware Interface (EFI) Інтерфейс між операційною системою і мікропрограмами, які керують низькорівневими функціями комп'ютерного обладнання. Основне призначення UEFI: коректно ініціалізувати обладнання при увімкненні системи та передати управління завантажувачу операційної системи. UEFI призначений для заміни BIOS — інтерфейсу, який традиційно використовується всіма IBM PC-сумісними персональними комп'ютерами. Перша специфікація UEFI (тоді ще просто "EFI") була розроблена компанією Intel, пізніше від першої назви відмовилися й остання версія стандарту має назву Unified Extensible Firmware Interface (UEFI). Наразі розробкою UEFI займається Unified EFI Forum.

USB Universal Serial Bus. Стандарт роз'ємів і кабелів для передачі даних (до 40 Гбіт/с) та живлення (до 240 Вт) невеликих пристроїв. Метою створення USB стандарту є:

- — краща уніфікація роз'ємів і кабелів;
- — нормування використання енергії;
- — створення протоколів обміну даними;
- — уніфікація функціональності і драйверів приладів;
- — можливість «гарячого» приєднання (та, як правило, і від'єднання) пристроїв.

.

VPN Virtual Private Network. Узагальнена назва технологій, які дозволяють створювати віртуальні захищені мережі поверх інших мереж із меншим рівнем довіри. VPN-тунель, який створюється між двома вузлами, дозволяє приєднаному пристрою чи користувачу бути повноцінним учасником віддале-

ної мережі і користуватись її сервісами — внутрішніми сайтами, базами, принтерами, політиками виходу в Інтернет. Безпека передавання інформації через загальнодоступні мережі реалізована за допомогою шифрування, внаслідок чого створюється закритий для сторонніх канал обміну інформацією. Технологія VPN дозволяє об'єднати декілька географічно віддалених мереж (або окремих клієнтів) в єдину мережу з використанням для зв'язку між ними непідконтрольних каналів. Багато провайдерів пропонують свої послуги як з організації VPN-мереж для бізнес-клієнтів, так і для виходу в мережу Інтернет. VPN є клієнт-серверною технологією.

ВСТУП

Нині, наше життя дуже тісно пов’язано з різноманітними технологіями. Ці технології мають найрізноманітніший вигляд. Деякі з них стали для нас буденними, як от телефони або Global Positioning System (GPS) чи каса в магазині. Інші ж вже представлені є досить диковинними: штучний інтелект, автопілот в авто, супутниковий інтернет. Хоч ці технології не дуже розповсюджені в нашому биту, але згодом вони також стануть невіддільною частиною буденного життя.

Але жодна з цих технологій не може працювати без однієї, найважливішої з технологій — мережі. Термін мережа має безліч значень в залежності від контексту використання. Але незмінним залишається її основне значення — поєднання чогось розрізненого в єдину систему. Так комп’ютерна мережа поєднує різні комп’ютери в єдину комп’ютерну систему. Так само й соціальні мережі поєднують людей у спільноти.

Хоч мережа повністю пронизує наше життя, але вона завжди залишається в тіні. Тому більшість людей зневажають її. Тим самим допускаючи фатальну помилку. Для прикладу, найбільша кількість злочинів у світі зараз відбувається через мережу. Починаючи від шпигунства і шахрайства і закінчуючи організацією терористичних актів.

Але чи є ці злочинці злочинцями? Хтось скаже так, хтось ні. Насправді це неоднозначне питання. Відповідь залежить від того з якої сторони дивитися на це питання. Адже ці злочинці такі самі як Робін Гуд. З однієї сторони він грабує людей, а з іншої допомагає людям. Так хто він злочинець чи герой? Тут та сама ситуація, але з іншими діями. З однієї сторони ці злочинці порушують своїми діями закони на норми. А іншої сторони спонукають людей до саморозвитку, навчання та поліпшення наявних технологій.

Зараз дуже мала частка людей поведеться на рекламу "Ви виграли 1 000 000 хоча ще 10 років тому велись. Також нині браузери мають цілий спектр захисних механізмів, що забезпечують безпечний перегляд вебсторінок. А різноманітні криптографічні протоколи, що постійно поліпшуються, захищають користувачі від перехоплення їх персональних даних.

Саме злочини, що виконуються через мережу, роблять її безпечнішою для користувачів. Але навіть так, це дуже небезпечне місце для недосвідченого користувача. Саме тому дуже важливо навчати людей користуватися мережею. Ко-

жний має знати, як захищатися в мережі від зловмисників. Але, на жаль, це вміють одиниці.

Хоч мережа і є небезпечним місцем, вона є найпотужнішою з усіх технологій. Саме через мережу відбувається комунікація людей, банківські платежі, перегляд відео та вебсайтів, працює пошта. І це крихітний перелік її можливостей. Мережа надає своїм користувачам неймовірні можливості. Тому її використання та подальше поліпшення є необхідним для людства в цілому. Але для цього мережу треба вивчати.

Зараз вищі навчальні заклади майже зовсім не використовують сучасні технології та можливості мережі в навчанні студентів. Тим самим залишаючи величезну безодню в знаннях студентів про мережу. Її використання в навчанні значно б полегшив і поліпшив якість навчання студентів. До того ж навчив їх користуватися мережею. Тим самим зменшивши можливості зловмисників.

Крім того вивчення мережі допоможе студентам в розумінні внутрішніх процесів та структури. Та, відповідно, надасть можливість та заохотить їх до створення нових технологій. Адже неможливо створити ядерний реактор без знання базової фізики. Використання мережевих технологій зробить прозорішим для студентів адміністративні процеси університету.

Особливо явною проблема стала протягом карантину та військових дій. Через відсутність повноцінної мережі та неможливість проводження занять в класичному режимі весь навчальний процес став майже паралізованим. Та походив не на навчання, а на посміховисько. У той час використання мережевих технологій значною мірою допоміг би максимально нейтралізувати цю проблему.

Потужна і правильно налаштована мережа допоможе не тільки студентам, а й працівникам. Вона забезпечить швидке та зручне середовище для комунікації, документообігу та безлічі інших адміністративних процесів. До того ж наявність мережевих служб значною мірою автоматизує всі адміністративні процеси. Пришвидшивши їх та зменшивши шанс виникнення помилки через людський фактор.

Саме тому ця робота посвячена саме поліпшенню поточної ситуації та запровадження сучасних технологій в кафедральну мережу. Цілю роботи не є побудова найпотужнішої та найзахищенішої мережі, а забезпечення хоча б базових, щонайнеобхідніших мережевих служб та послуг. В роботі представлено процес розробки та побудови простої, швидкої й, що найважливіше, максимально безпечної мережі.

1. ПРОПОЗИЦІЇ

1.1. Наявний стан кафедри та пропозиції

Нині на кафедрі наявні 53 комп'ютери та 1 сервер. Комп'ютери закупалися в різні роки та мають різні характеристики, тому наводитися не будуть.

Попри те, що найновіші комп'ютери закуплені о 2017 році. Вони мають дуже опосередковані характеристики через не раціональний підбір компонентів. Більшість всіх комп'ютерів кафедри є абсолютно застарілими та не відповідають жодним з сучасних вимог. Більша частина комп'ютерів кафедри керуються операційною системою Windows XP.

Наявний сервер також є абсолютно застарілим. Він має слабкий, за сучасним вимірами процесор, всього 2 гігабайти оперативної пам'яті та 160 гігабайтів дискового простору. Їм керує операційна система Windows Server 2003, що також є абсолютно застарілою за всіма параметрами.

Логічно та фізично мережа кафедри поділена на 2 мережі:

— адміністративна. До цієї мережі під'єднано всі комп'ютери персоналу кафедри та сервер. Вона напряму виходить за зону досяжності та керується стороннім персоналом Її адреса 192.168.0.0/24;
— студентська. До цієї мережі під'єднано всі студентські комп'ютери та сервер. Сервер виконує роль шлюзу для цієї мережі. Ця мережа повністю контролюється кафедральним сервером та не виходить за межі кафедри. Вона має адресу 10.0.0.0/24.

Всі проміжні вузли цих мереж є неконтрольованими комутаторами. Цю архітектуру мережі можна вважати досить розумною, але її захищеність викликає питання.

Поточна архітектура мережі наведена на рисунку 1.1.

У той час, програмне забезпечення кафедри потребує негайної модернізації.

1.2. Варіанти щодо поліпшення архітектури мережі

Для поліпшення архітектури мережі пропонується перенести всі адміністративні комп'ютери в мережу 10.0.0.0/24. Тим самим розмістивши їх за брандмауером. А мережу 192.168.0.0/24 використовувати як зовнішню мережу. Також пропонується додати Virtual Private Network (VPN) мережу з адресою 10.0.1.0/24 для

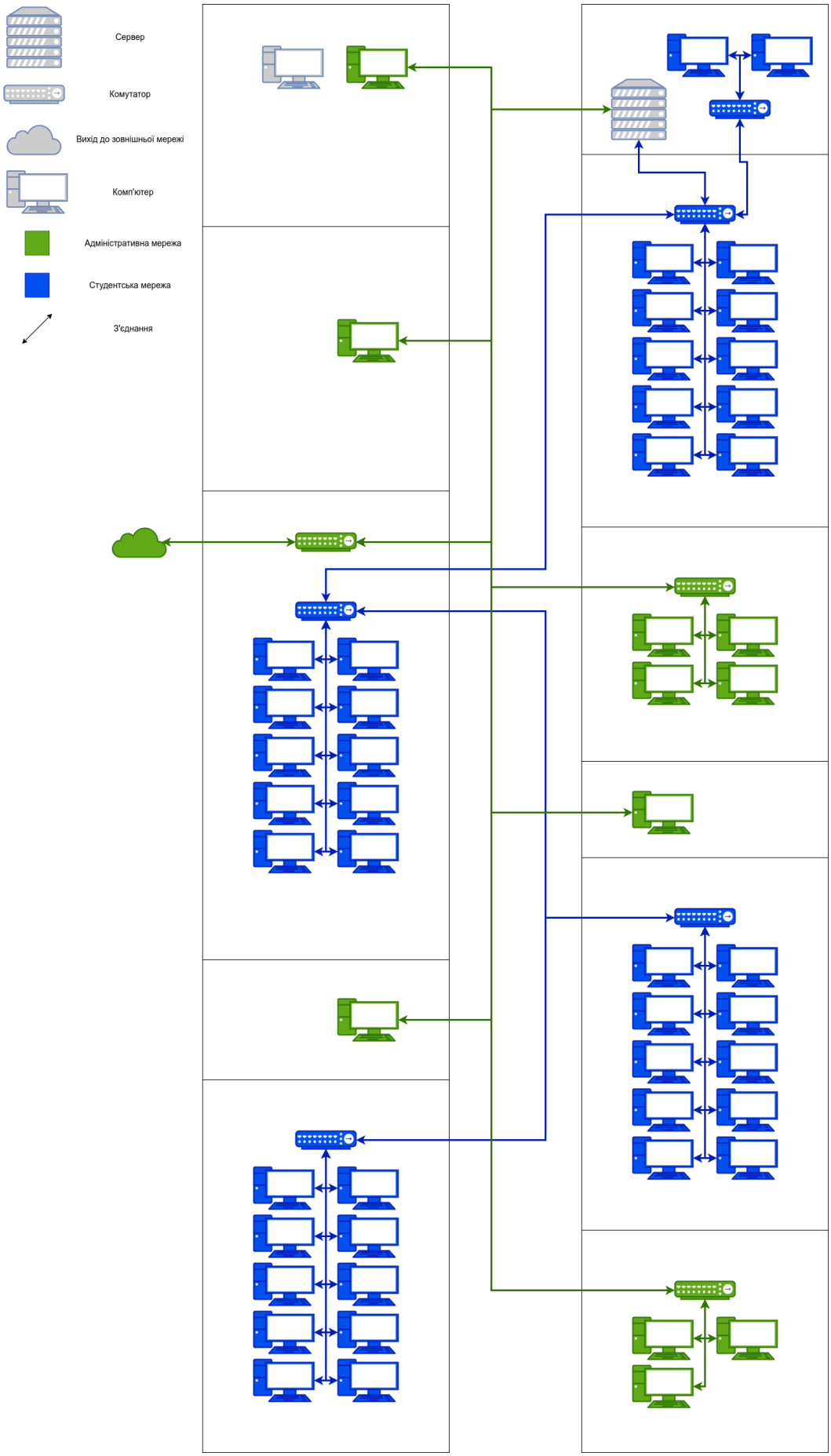

Рисунок 1.1 — Наявна мережа кафедри

запровадження віддаленого доступу до мережі кафедри.

Запропонована архітектура мережі наведена на рисунку 1.2.

Пропонується запровадити моногамну архітектуру мережі. Це дозволить значно спростити програмний стек мережі та підвищити її надійність. Для мережі пропонується запровадити захист за моделлю Self Defence. Це дозволить значно поліпшити захист мережі не змінюючи всі проміжні комутатори мережі.

Пропонується замінити поточний сервер на сервер SuperMicro. Сервер SuperMicro наявний на кафедрі, але жодним чином не використовується. У порівнянні з поточним сервером він має значно ліпші характеристики.

Також пропонується запровадити в мережі кафедри ряд мережевих сервісів:

— DHCP. Цей сервіс дозволить автоматизувати налаштування всіх комп'ютерів кафедри. Також сприятиме автоматичному оновленню параметрів. Завдяки чому кожний компонент мережі матимуть актуальні налаштування мережі. А нові компоненти автоматично інтегруватимуться в мережу. Не потребуючи будь-яких ручних налаштувань;
— DNS. Цей сервіс призначений в першу чергу для підвищення зручності користуванням мережею та її сервісами. Доменні імена значно легше використовувати ніж мережеві адреси. До того ж адреси тих чи інших сервісів чи компонентів мережі можуть змінюватися. Це може викликати плутанину та необхідність ручного втручання. Використання власного доменного сервера також надає ряд інших переваг: власний простір імен, пришвидшення обробки запитів, незалежність від постачальників послуг тощо;
— DDNS. Це допоміжний сервіс призначений для автоматичної реєстрації або оновлення доменних імен. Таким чином кожний компонент мережі автоматично отримуватиме власне доменне ім'я. Він отримує відомості від DHCP сервера та направляє відповідні запити до DNS сервера. Забезпечуючи синхронізацію цих окремих сервісів;
— Single Sign-On (SSO). Дуже складний в налаштуванні сервіс, але не має аналогів. Він забезпечує централізовану ідентифікацію, автентифікацію та керування користувачами. Завдяки йому користувач може автентифікуватися на будь-якому комп'ютері мережі та отримати повний доступ до свого робочого простору. Також цей сервіс дозволяє виконувати лице одну автентифікацію користувачеві. Передаючи данні про нього іншим сервісам (напри-

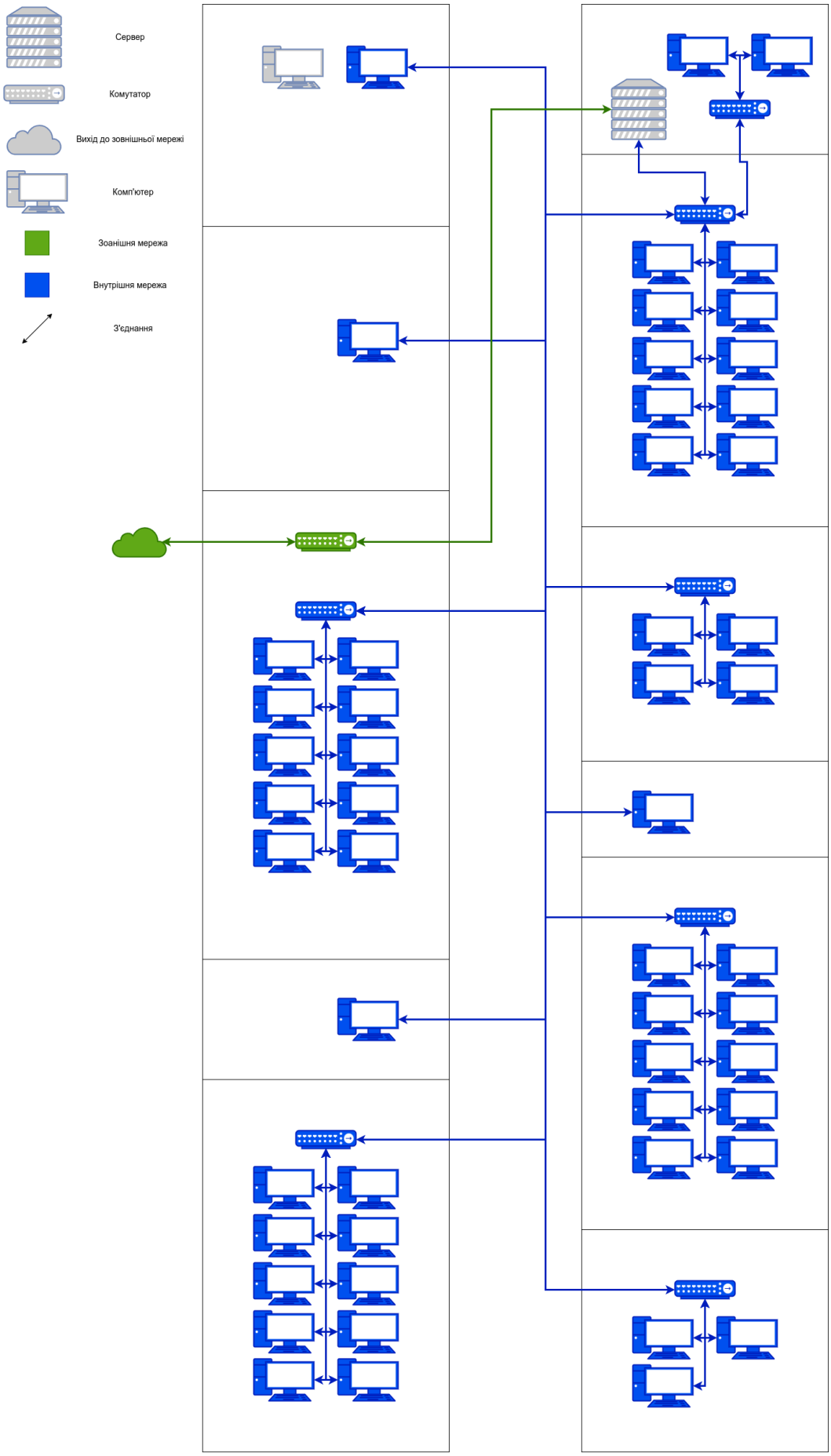

Рисунок 1.2 — Запропонована мережа кафедри

клад, пошта).

Найнеобхіднішим є позбавлення від застарілого програмного забезпечення, а особливо операційних систем.

1.3. Обрання сімейства операційних систем

1.3.1. Загальна інформація

У наш час існує досить багато сімейств операційних систем і ще більше самих систем. Поняття сімейства є дуже розпливчатим і здебільшого залежить від методології та критеріїв класифікації. В найпростішому вигляді існує 2 основні активні сімейства Windows та Unix-like.

До сімейства Windows відносяться тільки операційні системи Windows розроблені компанією Microsoft.

Сімейство Unix-like, своєю чергою, дуже велике та дуже активно розвивається. Це сімейство відоме своєю надійністю. Linux, Berkeley Software Distribution (BSD) і MacOS є найвідомішими сімействами у складі Unix-like.

Кожне з перелічених сімейств дуже активно використовуються та займаю лідерство у своїй сфері. Windows та MacOS є лідерами ринку операційних систем для користувацьких комп'ютерів. Своєю чергою Linux і BSD є лідерами у сфері серверних систем. Сімейство Linux також є абсолютним лідером у сфері мікрокомп'ютерів, роутерів, комутаторів, телефонів та Internet of Things (IoT).

Яскравими представниками сімейства BSD є:

— FreeBSD;
— OpenBSD;
— NetBSD;
— OpenSolaris.

Яскравими представниками сімейства Linux є:

— Debian;
— RedHat Enterprise Linux (RHEL);
— Ubuntu;
— Android;
— CentOS;
— Fedora;
— OpenSuse;

— Arch.

Генологічне дерево сімейства Unix-like наведено на рисунку 1.3, котре запозичено з енциклопедії [1].

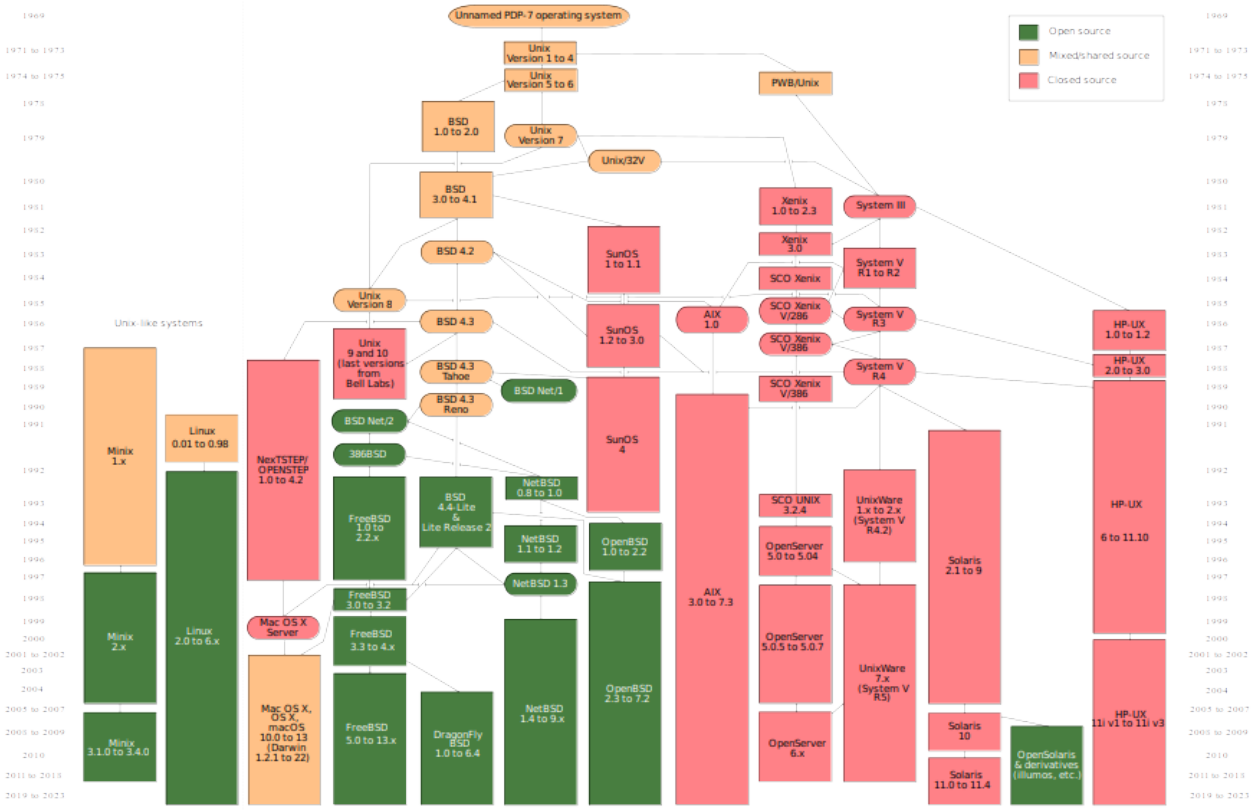

Рисунок 1.3 — Генологічне дерево сімейства Unix-like

Сімейство Linux є, мабуть, найрозвиненішим з усіх сімейств у світі. Його генологічне дерево, станом на 2023, наведено у рисунку [2].

Сімейство MacOS поставляється тільки з пристроями компанії Apple, тому розглядатися не буде. Сімейство BSD також не розглядається як варіант через свою спеціалізованість та специфічність.

Таким чином для огляду обрано два сімейства: Windows та Linux. Опис сімейства операційних систем Windows надано у додатку А. Опис сімейства операційних систем Linux надано у додатку Б.

1.3.2. Обрання дистрибутиву

Проаналізувавши різноманітні дистрибутиви сімейства Linux обрано дистрибутив Manjaro.

Цей дистрибутив надає користувачам найновіші версії програмного забезпечення. Оновлення системи та користувацьких програм відбувається за моделлю

Rolling. Завдяки чому система завжди залишається в актуальному стані та не потребує періодичних оновлень чи перевстановлення.

Дистрибутив використовує пакетний менеджер Pacman. Особливостями цього менеджера є модульність, завдяки чому можна встановлювати лише необхідні компоненти програмного забезпечення. Він має величезний набір пакетів завдяки використанню Arch User Repository (AUR). Тобто, Pacman має два основні джерела пакетів:

— офіційні репозиторії. Ці репозиторії керується розробниками та надають максимально надійні версії пакетів. З гарантованою сумісністю версій;
— AUR. Цей репозиторій керується користувачами. Він надає величезний набір пакетів. Але ці пакети можуть бути несумісними з поточними версіями програм та бібліотек. Тому його необхідно використовувати з обережністю.

1.4. Висновки за розділом

Виходячи з перелічених вище факторів та описів сімейств операційних систем (див. додатки А та Б) прийнято рішення про використання на комп'ютерах та сервері кафедри операційної системи сімейства Linux. Серед дистрибутивів Linux обрано дистрибутив Manjaro.

2. ВСТАНОВЛЕННЯ ОПЕРАЦІЙНОЇ СИСТЕМИ

2.1. Завантаження з носія адміністратора

Для встановлення операційної системи на сервер вставляємо носій адміністратора в Universal Serial Bus (USB) порт сервера. Повний опис процесу створення носія адміністратора наведено у додатку В. Під час завантаження сервера викликаємо меню обрання носія завантаження. В переліку обираємо носій адміністратора. Сервер підтримує Unified Extensible Firmware Interface (UEFI), тому завантажуємось саме в цьому режимі.

Після завантаження носія вводимо ім'я та пароль адміністратора.

Процес збірки та встановлення системи базується на записі розробників дистрибутиву з форуму Manjaro [3].

2.2. Налаштування тимчасової мережі

Для перевірки поточних конфігурацій мережі використовуємо утиліту ip [4].

```
[root@admin-usb ~]# ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
↪  qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eno1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
↪  default qlen 1000
    link/ether ac:1f:6b:01:04:1a brd ff:ff:ff:ff:ff:ff
    altname enp4s0f0
    inet6 fe80::ae1f:6bff:fe01:41a/64 scope link
       valid_lft forever preferred_lft forever
3: eno2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
↪  default qlen 1000
    link/ether ac:1f:6b:01:04:1b brd ff:ff:ff:ff:ff:ff
    altname enp4s0f1
    inet6 fe80::ae1f:6bff:fe01:41b/64 scope link
       valid_lft forever preferred_lft forever
[root@admin-usb ~]# ip route
```

З виводу команд бачимо, що наразі жодний з інтерфейсів не налаштований і відсутні будь-які маршрути. Це зовсім очікувана поведінка. Попри налаштований

на носії DHCP клієнт, він нічого не налаштував. Це відбулось через відсутність в мережі DHCP сервера. Тому необхідну налаштувати мережу власноруч.

Утиліта ip [4] вивела 3 інтерфейси:

— lo — петлевий інтерфейс. Цей інтерфейс завжди має адресу 127.0.0.1 і використовується тільки для локального доступу (звернення до самого себе);
— eno1 — інтерфейс під'єднаний до зовнішньої мережі. Цей інтерфейс використовується для доступу у зовнішню мережу і, відповідно, до мережі інтернет;
— eno2 — інтерфейс під'єднаний до внутрішньої мережі. Цей інтерфейс використовується для зв'язку з хостами у внутрішній (захищеній) мережі.

Налаштуємо інтерфейс eno1. Для початку вказуємо йому ip адресу з пулу зовнішньої мережі. Використаємо стару адресу сервера, тобто 192.168.0.100/24.

```
[root@admin-usb ~]# ip address add 192.168.0.100/24 dev eno1
```

Також додаємо маршрут за замовчуванням до шлюзу. Це необхідно для отримання доступу до зовнішньої мережі та мережі інтернет.

```
[root@admin-usb ~]# ip route add default via 192.168.0.1
```

Тепер налаштовуємо інтерфейс eno2. Для цього достатньо лише видати адресу. Обрана адреса для інтерфейсу — 10.0.0.1/24.

```
[root@admin-usb ~]# ip address add 10.0.0.1/24 dev eno2
```

Перевіряємо внесені налаштування.

```
[root@admin-usb ~]# ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
↪  qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eno1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
↪  default qlen 1000
    link/ether ac:1f:6b:01:04:1a brd ff:ff:ff:ff:ff:ff
    altname enp4s0f0
    inet 192.168.0.100/24 scope global eno1
```

```
       valid_lft forever preferred_lft forever
    inet6 fe80::ae1f:6bff:fe01:41a/64 scope link
       valid_lft forever preferred_lft forever
3: eno2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
↪  default qlen 1000
    link/ether ac:1f:6b:01:04:1b brd ff:ff:ff:ff:ff:ff
    altname enp4s0f1
    inet 10.0.0.1/24 scope global eno2
       valid_lft forever preferred_lft forever
    inet6 fe80::ae1f:6bff:fe01:41b/64 scope link
       valid_lft forever preferred_lft forever
[root@admin-usb ~]# ip route
default via 192.168.0.1 dev eno1
10.0.0.0/24 dev eno2 proto kernel scope link src 10.0.0.1
192.168.0.0/24 dev eno1 proto kernel scope link src 192.168.0.100
```

Також необхідно власноруч налаштувати DNS розв'язувач. Для цього відредагуємо файл /etc/resolv.conf [5], додавши в нього відомості про DNS сервер.

```
[root@admin-usb ~]# echo "nameserver 1.1.1.1" >> /etc/resolv.conf
```

Перевіряємо працездатність за допомогою утиліти ping [6]. Наявність зв’язку продемонструє коректність налаштувань мережі. Використання доменного імені в запиті продемонструє коректність роботи DNS клієнта та наявність його зв’язку з DNS сервером.

```
[root@admin-usb ~]# ping -c 10 google.com
ЛУНА-ІМПУЛЬС google.com (216.58.215.110) 56(84) байтів даних.
64 байтів з waw02s17-in-f14.1e100.net (216.58.215.110): icmp_seq=1 ttl=111
↪  час=55.3 мс
64 байтів з waw02s17-in-f14.1e100.net (216.58.215.110): icmp_seq=2 ttl=111
↪  час=56.0 мс
64 байтів з waw02s17-in-f14.1e100.net (216.58.215.110): icmp_seq=3 ttl=111
↪  час=55.3 мс
64 байтів з waw02s17-in-f14.1e100.net (216.58.215.110): icmp_seq=4 ttl=111
↪  час=73.8 мс
64 байтів з waw02s17-in-f14.1e100.net (216.58.215.110): icmp_seq=5 ttl=111
↪  час=47.8 мс
64 байтів з waw02s17-in-f14.1e100.net (216.58.215.110): icmp_seq=6 ttl=111
↪  час=70.6 мс
64 байтів з waw02s17-in-f14.1e100.net (216.58.215.110): icmp_seq=7 ttl=111
↪  час=68.5 мс
64 байтів з waw02s17-in-f14.1e100.net (216.58.215.110): icmp_seq=8 ttl=111
↪  час=67.1 мс
64 байтів з waw02s17-in-f14.1e100.net (216.58.215.110): icmp_seq=9 ttl=111
↪  час=66.8 мс
```

```
64 байтів з waw02s17-in-f14.1e100.net (216.58.215.110): icmp_seq=10 ttl=111
 ↪  час=51.4 мс

--- Статистика луна-імпульсів google.com ---
Передано 10 пакетів, отримано 10, 0% packet loss, time 9013ms
rtt min/avg/max/mdev = 47.801/61.262/73.845/8.609 ms
```

Вивід демонструє коректність роботи мережі та наявність доступу до мережі інтернет. Тепер налаштувавши відповідні параметри мережі на комп'ютері можемо під'єднатися до сервера.

2.3. Налаштування віддаленого доступу

Для під'єднання до сервера необхідно налаштувати ssh [7] клієнт на комп'ютері. Для цього редагуємо файл /.ssh/config [8] і додаємо в нього наступну конфігурацію.

```
Match host admin-usb exec "knock -d 100 10.0.0.1 0 {ТАЄМНА_ПОСЛІДОВНІСТЬ}"
Host  admin-usb
    Hostname 10.0.0.1
    User root
    Port 22
```

Ця конфігурація реагуватиме на хост admin-usb. Перед з'єднанням з сервером через SSH, клієнт виконує ввід таємної послідовності за допомогою програми knock [9]. Послідовність починається з 0. Це число не входить до таємної послідовності та призведе до скидання послідовності. Етапи послідовності надсилається з проміжком 100 мілісекунд. Цей час надасть серверу певний час на обробку запиту, зменшивши вірогідність виникнення випадкових. Поле Hostname містить, тимчасово, поточну внутрішню адресу сервера.

Під'єднуємось до сервера за допомогою SSH клієнта.

```
[root@hostname ~]# ssh admin-usb.eom
```

Подальші налаштування проводяться за допомогою віддаленого доступу через SSH.

2.4. Розподілення дискового простору

Ідентифікуємо наявне дискове обладнання сервера. Для виведення необхідної інформації використовуємо програму lsblk [10].

```
[root@admin-usb ~]# lsblk --noempty --scsi
NAME HCTL       TYPE VENDOR   MODEL           REV SERIAL                          TRAN
sda  8:0:0:0    disk ATA      SuperMicro SSD  0R   SMC0515D95216CNG3116
↪  sata
sdb  10:0:0:0   disk          USB FLASH DRIVE PMAP 07010B32CF82E347
↪  usb
sdc  11:2:0:0   disk AVAGO    SMC3108         4.65 00be9b351d8e6c6a2a02b6021a800403
```

З виводу бачимо наявність 3-ох дискових накопичувачів:

— sda — вбудований Solid-State Drive (SSD) накопичувач під'єднаний через шину Serial AT Attachment (SATA);

— sdb — USB накопичувач адміністратора;

— sdc — логічний накопичувач згенерований вбудованим Redundant Array of Independent Disks (RAID) контролером.

Схема запланованого розподілу дискового простору сервера наведено у таблиці 2.1.

Таблиця 2.1 — Схема розподілу дискового простору сервера

Диск	Розділ	Тип	Розмір	Файлова система	Точка монтування
sda	1	EFI System	512 MiB	fat32	/efi
	2	Linux filesystem	Весь доступний простір	ext4	/
sdb	1	Linux swap	128 GiB	swap	swap
	2	Linux filesystem	Весь доступний простір	ext4	/var

Дисковому просту RAID масиву призначено точку монтування /var, виходячи зі специфіки сервера та його завдань. Відповідно до Filesystem Hierarchy Standard (FHS) [11] в директорії /var знаходяться всі робочі дані програм та служб. У той

час, директорія /home, в якій зберігаються дані користувачів, на сервері активно використовуватися не буде.

Розподілення дискового простору відбувається за допомогою утиліти fdisk [12]. Інструкція з використання програми наведена у додатку Г. Результат розподілення дискового простору перевіряємо за допомогою команди lsblk [10], з ключем noempty.

```
[root@admin-usb ~]# lsblk --noempty
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda      8:0    0   59G  0 disk
├─sda1   8:1    0  512M  0 part
└─sda2   8:2    0 58,5G  0 part
sdb      8:16   1 28,9G  0 disk
├─sdb1   8:17   1    8M  0 part
├─sdb2   8:18   1  512M  0 part /efi
└─sdb3   8:19   1 28,4G  0 part /
sdc      8:32   0 18,2T  0 disk
├─sdc1   8:33   0  128G  0 part
└─sdc2   8:34   0 18,1T  0 part
```

Далі ініціалізуємо заплановані файлові системи. Інструкція з ініціалізації файлових систем наведена у додатку Д.

Структурований результат розподілу та форматування дискового простору наведений у таблиці 2.2.

Таблиця 2.2 — Схема розподілу дискового простору сервера

Диск	Розділ	Ідентифікатор	Файлова система	Точка монтування
sda	1	0879-0B5E	fat32	/efi
	2	be48508f-45a1-462c-85d8-368c7086a657	ext4	/
sdb	1	6fc4811b-7cb2-48ab-81c7-fe95284fb9e8	swap	swap
	2	cd4b8675-362e-4874-8dec-ea7692946e27m	ext4	/var

2.5. Ініціалізація віртуального середовища

Під’єднуємо створені файлові системи відповідно до схеми у таблиці 2.2 до директорії /mnt/target.

```
[root@admin-usb ~]# mount --mkdir /dev/sda2 /mnt/target
[root@admin-usb ~]# mount --mkdir /dev/sda1 /mnt/target/efi
[root@admin-usb ~]# mount --mkdir /dev/sdc2 /mnt/target/var
```

Ініціалізуємо базовий образ системи в точці монтування.

```
[root@admin-usb ~]# basestrap /mnt/target/ base
```

Переходимо у віртуальне середовище нової системи.

```
[root@admin-usb ~]# manjaro-chroot /mnt/target/ /bin/bash
```

Оновлюємо бази даних індексів пакетів менеджера pacman [13; 14].

```
[root@admin-usb /]# pacman -Syy
```

2.6. Встановлення мінімально необхідного набору пакетів

Спочатку необхідно встановити необхідні для працездатності системи пакети:

- linux61. Ядро Linux 6.1;
- linux-firmware. Набір мікропрограм низького рівня (драйверів);
- grub. Завантажувач операційної системи;
- grub-theme-manjaro. Тема для завантажувача від розробників дистрибутиву;
- efibootmgr. Програма для адміністрування UEFI Boot Manager;
- mkinitcpio. Утиліта для генерації initramfs образів;
- mdadm. Система для взаємодії з RAID масивами;
- manjaro-release. Містить інформацію про поточний реліз системи та надає його програмам при запиті;
- nano. Текстовий редактор.

В сервері встановлено RAID масив. Деякі контролери RAID повністю абстрагують систему від керування. Інші ж цього не роблять. Тому дуже бажано встановити та налаштувати сервіс mdadm [15] під час завантаження, про всяк випадок. Це гарантує коректність завантаження системи в обох випадках.

Для встановлення перелічених вище пакетів виконуємо наступну команду.

```
[root@admin-usb /]# pacman -Syyu linux61 linux-firmware grub grub-theme-manjaro
↪  efibootmgr mkinitcpio mdadm manjaro-release nano
```

2.7. Налаштування системи завантаження

2.7.1. Опис процесу завантаження системи

Процес завантаження систем Linux можна побілити на 3 основні етапи:

— завантаження завантажувача;
— завантаження островів ядра та системи;
— завантаження системи.

Першою чергою, при завантаженні комп'ютера, викликається система UEFI або, у старіших системах, Basic Input/Output System (BIOS). Головною задачею цієї системи, в даному контексті, пошук та завантаження завантажувача в оперативну пам'ять. Далі завантажувачу надається повний контроль над системою.

Завантажувач відповідає та завантаження острова ядра й острова системи в оперативну пам'ять. Деякі завантажувачі надають користувачеві вибір операційної системи для завантаження. Після завантаження островів завантажувач передає керування острову ядра.

Острів ядра — це мінімальний образ ядра. Першим і найголовнішим завданням острова є завантаження модулів ядра. Таким чином острів перетворюється на повноцінне ядро.

Острів системи — це мінімальний образ системи. Він використовується ядром, як контекст. Острів володіє наймінімальнішим набором програм та конфігурацій, необхідний для завантаження основної системи.

Після ініціалізації необхідної апаратної частини та монтування файлових систем. Острів системи замінюється на основну систему. Після чого відбувається фінальний етап завантаження — ініціалізація основної системи.

Комусь цей процес може здатися надлишковим. Але саме цей варіант є максимально надійним, адаптивним та відмовостійким. До того ж у випадку виникнення помилки система залишиться у стані острова. В цей момент керування надається користувачу, надаючи можливість усунення несправності.

2.7.2. Налаштування системи збірки

Для генерації островів використовуватимемо систему збірки mkinitcpio [16]. Її налаштування відбувається шляхом редагування файл конфігурації. Цей файл розміщений за шляхом /etc/mkinitcpio.conf [17].

Для забезпечення підтримки RAID, додаємо гачок mdadm_udev після гачка udev. Також для активації можливості гібернації системи, додаємо гачок resume після гачка mdadm_udev.

Вмикаємо стиснення островів алгоритмом gzip, призначивши параметру COMPRESSION значення gzip. І вимикаємо розархівування модулів ядра, надавши параметру MODULES_DECOMPRESS значення no.

Повний файл конфігурації наведено у додатку Е.

2.7.3. Налаштування завантажувача

Для завантаження системи використовуватимемо завантажувач Grand Unified Bootloader 2 (GRUB) [18]. Для його налаштування відредагуємо файл /etc/default/grub.

За замовчуванням завантажувач одразу намагається завантажити систему. Тому для надання можливості змінити його поведінку при завантаженні додаємо затримку завантаження довжиною 5 секунд. Як поведінку затримки обрано варіант menu.

```
GRUB_TIMEOUT=5
GRUB_TIMEOUT_STYLE=menu
```

Для запобігання плутанини та генерації непотрібних пунктів меню завантажувача вимкнено автоматичний пошук встановлених систем.

```
GRUB_DISABLE_OS_PROBER=true
```

Додаємо інформацію про розділ swap в аргументи ядра. Це необхідно до забезпечення виходу системи з гібернації та відновлення роботи.

```
GRUB_CMDLINE_LINUX_DEFAULT="resume=UUID=6fc4811b-7cb2-48ab-81c7-fe95284fb9e8
↪  quiet udev.log_priority=3"
```

Також змінено параметр GRUB_DISABLE_RECOVERY. Завдяки цьому завантажувач будуватиме 2 типи ядра: звичайне та резервне. Це необхідно на випадок виникнення помилки та надання можливості завантажитись у безпечному режимі. Що своєю чергою значно полегшить виправлення помилки.

```
GRUB_DISABLE_RECOVERY=false
```

Повний відредагований файл конфігурацій наведено у додатку Ж.

2.7.4. Налаштування автоматичного монтування файлових систем

Файл /etc/fstab [19] це конфігураційний файл системи де містяться відомості про пристрої та їх точки монтування, котрі необхідно змонтувати при ініціалізації системи. Це дуже важливий файл і вимагає додаткової уваги. Адже невірна конфігурація чи помилка в одній літері може призвести до неможливості завантаження системи.

Його синтаксис побудований у вигляді таблиці з шістьма колонками:

1. ідентифікатор;
2. точка монтування;
3. файлова система;
4. параметри монтування;
5. необхідність dump;
6. пріоритет перевірки файлової системи.

Використовуючи дані про форматування дискового простору з таблиці 2.2, створюємо відповідні записи у файлі /etc/fstab [19].

Готовий вміст файлу /etc/fstab [19] наведено у додатку И.

2.7.5. Встановлення завантажувача

Тепер встановімо завантажувач GRUB. Скориставшись програмою grub-install [20]. Встановлюємо завантажувач для систем UEFI за допомогою команди.

```
[root@admin-usb /]# grub-install --target=x86_64-efi  --efi-directory=/efi
↪  /dev/sda
```

Тепер будуємо завантажувальний образ системи та острів ядра системи. Також оновлюємо конфігурації завантажувача. Для генерації образу скористаємося системою збірки mkinitcpio [16], котру вже налаштовано. Для генерації образу виконуємо програму mkinitcpio [16] з ключем allpresets.

```
[root@admin-usb /]# mkinitcpio --allpresets
```

Оновлюємо конфігурації завантажувача за допомогою утиліти update-grub.

```
[root@admin-usb /]# update-grub
```

2.8. Налаштування мережі

Сервер має статичні налаштування мережі, тому DHCP клієнт використовуватися не буде. Сервер містить 2 фізичні порти, котрі пов’язані з файлами eno1 та eno2.

Інтерфейс eno1 під’єднано до зовнішньої мережі з адресою 192.168.0.0/24. Його ip адреса в цій мережі має бути 192.168.0.100/24. Шлюз мережі знаходиться за адресою 192.168.0.1/24.

Для налаштування цього інтерфейсу створюємо файл eno1.network [21] в директорії конфігурації systemd-networkd [22]. Вміст файлу /etc/systemd/network/eno1.network наведено нижче.

```
[Match]
Name=eno1

[Network]
Description=External network
Address=192.168.0.100/24
Gateway=192.168.0.1
```

Інтерфейс eno2 під’єднано до внутрішньої мережі з адресою 10.0.0.0/24. Його ip адреса в цій мережі має бути 10.0.0.1/24. Роль шлюзу мережі виконує сам сервер, тому цей параметр не вказуємо.

Для налаштування цього інтерфейсу створімо файл eno2.network [21] в директорії конфігурації systemd-networkd [22]. Вміст файлу /etc/systemd/network/eno2.network наведено нижче.

```
[Match]
Name=eno2

[Network]
Description=Internal network
Address=10.0.0.1/24
```

Активуємо службу systemd-networkd [22] за допомогою наступної команди.

```
[root@admin-usb /]# systemctl enable systemd-networkd.service
```

2.9. Налаштування брандмауера

Брандмауер — це найголовніший рубіж захисту сервера від зловмисників. Його налаштування вимагають особливої уваги та ретельності. Для захисту сервера обрано потужний брандмауер NetFilter [23] з конфігуратором nftables [24] Налаштування брандмауера відбувається шляхом редагування файлу /etc/nftables.conf [24].

В конфігурації пишемо правила, за якими брандмауер фільтруватиме трафік. Фреймворк nftables [24] має можливість збереження стану. Завдяки цьому можемо реалізувати port knocking засобами nftables [24].

Основна мета port knocking — запобігти скануванню зловмисником системи на наявність потенційно придатних до зламу служб за допомогою сканування портів. Якщо зловмисник не надішле правильну послідовність, захищені порти виглядатимуть закритими [25].

Port knocking це дуже ефективний допоміжний захисний метод для приховування чутливих або не потрібних для загального використання портів. Доступ до них надається лише після вводу таємної послідовності. У випадку nftables доступ надається тільки користувачам, що виконали таємну послідовність.

Ця реалізація надає користувачеві 10 секунд на створення з'єднання з захищеними портами. 1 секунда надається для переходу між етапами послідовності. В разі помилки на будь-якому етапі або завершення часу очікування послідовність повністю скидається. Також брандмауер жодним чином не повідомляє про будь-які події. Тим самим значно ускладнюючи процес підбору послідовності.

Середній простір атаки на послідовність з 5 портів складає 79 бітів. В сучасних реаліях це не великий захист, але для допоміжного захисту досить. Своєю чергою послідовність довжиною 10 портів моє середній простір атаки 159 бітів. Також цей показних можна покращити, завдяки використанню різних протоколів. Але цей метод не надає великого приросту середнього простору атаки, у порівнянні зі збільшенням послідовності.

Наведена конфігурація (див. додаток К) забезпечує базові правила для захисту системи. У файлі конфігурації, в цілях безпеки, номери портів, що використовуються для таємної послідовності приховані під конструкцією {ВИПАДКОВИЙ_ПОРТ}. Ці правила не підходять для правильної роботи сервера та його служб. Тому будуть змінені пізніше.

Активуємо брандмауер та його автоматичне конфігурування запуском відповідної служби.

```
[root@admin-usb /]# systemctl enable nftables.service
```

2.10. Встановлення пароля адміністратора

Пароль це основний і базовий метод авторизації користувача в системах Linux. Доступ до сервера відбуватиметься за допомогою SSH (див. додаток Н), але пароль необхідний, як резервний варіант. Пароль адміністратора має бути надійним і зберігатися в надійному місці.

Для генерації пароля використовуємо команду.

```
[root@admin-usb /]# echo "$(tr -dc '[:graph:]' < /dev/urandom | head -c${1:-32})";
```

На вихід команда видасть нам згенерований пароль. Пароль складається з 32 печатних символів окрім пробілів. Середній простір атаки такого пароля складає 208 бітів, що є досить непоганим показником. Конкретно обраний варіант пароля має ентропію 184.12 біт, за оцінкою генератора паролів KeePassXC. Перелічені показники демонструють надійність обраного пароля.

Для зміни пароля користувача використовуємо утиліту passwd [26].

```
[root@admin-usb /]# passwd
```

2.11. Завершення

Також налаштовуємо додаткові, але не менш важливі параметри системи:

— змінюємо ім'я хосту на server. Відповідно до додатка Л;
— налаштовуємо локаль системи. Відповідно до додатка М;
— налаштовуємо віддалений доступ. Відповідно до додатка Н

Тепер виходимо з віртуального середовища.

```
[root@admin-usb /]# exit
```

Перезавантажуємо систему.

```
[root@admin-usb ~]# reboots
```

Після успішного завантаження системи переконуємось у коректності налаштувань системи.

Налаштуємо конфігурацію ssh для підключення до сервера. Для цього додамо нову конфігурацію до файлу /.ssh/config [8].

```
Match host server exec "knock -d 100 10.0.0.1 0 {ТАЄМНА_ПОСЛІДОВНІСТЬ}"
Host server
    Hostname 10.0.0.1
    User root
    Port 22
```

Під’єднуємось до сервера.

```
[root@admin-usb ~]# ssh server
```

Після успішної ідентифікації та автентифікації впевнюємось в коректності мережевих налаштувань та налаштувань сервісу sshd [27].

3. НАЛАШТУВАННЯ СЕРВІСІВ І БЕЗПЕКИ

3.1. Синхронізація часу

3.1.1. Загальні відомості

NTP — мережевий протокол синхронізації внутрішнього годинника комп'ютера з використанням мереж зі змінною латентністю, заснований на комутації пакетів.

Хоча традиційно NTP використовує для своєї роботи протокол UDP, він також здатний працювати й поверх TCP. Система NTP надзвичайно стійка до змін латентності середовища передачі.

NTP використовує алгоритм Марзулло (запропонований Кейтом Марзулло (Keith Marzullo) з Університету Каліфорнії, Сан-Дієго), включаючи таку особливість, як облік часу передачі. У версії 4 здатний досягати точності 10 мс (1/100 с) при роботі через Інтернет, і до 200 мікросекунд (1/5000 с) і краще усередині локальних мереж [28].

При створенні мережі дуже важливим аспектом є час. Безліч протоколів, особливо пов’язані з захистом, дуже чутливі до часу. Різниця в одну хвилину в деяких випадках є критичною. Тому дуже важливим є синхронізація часу в мережі. Кожний комп’ютер може бути налаштований на синхронізацію часу з мережі Інтернет. Це досить гарний варіант, але не для комплексної мережі.

Завдяки цьому налаштуванню NTP сервера мережа стане значно стабільнішою. При наявності доступу до мережі Інтернет сервер та клієнти будуть синхронізуватися з глобальними UTC серверами. У випадку відсутності доступу до інтернету сервер буде підтримувати мережу в синхронізованому стані, хоч і з певно похибкою. При відновленні доступу всі елементи мережі синхронізуються з глобальним часом.

Існує багато реалізацій цього протоколу. Серед них еталоном вважається ntpd. Цей демон має найповнішу реалізацію NTP. До того ж ntpd працює у режимах клієнта та сервера одночасно.

3.1.2. Встановлення та налаштування

Встановлюємо відповідний пакет.

```
[root@server ~]# pacman -Syyu ntp
```

Тепер відредагуємо його конфігураційний файл /etc/ntp.conf. Його відредагований вміст наведено у додатку П.

Ця конфігурація призначена для клієнт-серверної поведінки демона ntpd. ntpd при наявності з'єднання синхронізується з пулом серверів на території України та глобальним пулом. Таким чином демон може обрати оптимальні варіанти для синхронізації. Також він надаватиме послуги з синхронізації своїм клієнтам, але це відбудеться тільки при відсутності доступу клієнтів до жодного з інших серверів.

При наявності декількох серверів можна налаштувати їх самосинхронізацію. Через відсутність відповідних серверів ця можливість не налаштована і поки ігнорується.

Активуємо сервіс синхронізації часу.

```
[root@server ~]# systemctl enable --now ntpd
```

Перевіряємо наявність з'єднання з налаштованими серверами.

```
[root@server ~]# ntpq -p
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
 0.ua.pool.ntp.o .POOL.          16 p    -   64    0    0.000   +0.000   0.000
 1.ua.pool.ntp.o .POOL.          16 p    -   64    0    0.000   +0.000   0.000
 2.ua.pool.ntp.o .POOL.          16 p    -   64    0    0.000   +0.000   0.000
 3.ua.pool.ntp.o .POOL.          16 p    -   64    0    0.000   +0.000   0.000
 0.pool.ntp.org  .POOL.          16 p    -   64    0    0.000   +0.000   0.000
 1.pool.ntp.org  .POOL.          16 p    -   64    0    0.000   +0.000   0.000
 2.pool.ntp.org  .POOL.          16 p    -   64    0    0.000   +0.000   0.000
 3.pool.ntp.org  .POOL.          16 p    -   64    0    0.000   +0.000   0.000
```

Як бачимо, все працює правильно і синхронізація часу налаштована.

3.2. Система контейнеризації

3.2.1. Загальні відомості

Контейнеризація (віртуалізація на рівні операційної системи) — метод віртуалізації, при якому ядро операційної системи підтримує декілька ізольованих примірників простору користувача, замість одного. Ці примірники (часто звані

контейнерами або зонами) з точки зору користувача повністю ідентичні реальному серверові. Ядро забезпечує повну ізольованість контейнерів, тому програми з різних контейнерів не можуть впливати одна на одну [29].

Ця технологія є дуже якісною альтернативою для віртуалізації. Вона працює значно швидше та вимагає значно менше накладних ресурсів. Це досягається завдяки відсутності необхідності емуляції апаратних ресурсів.

Зараз ця технологія дуже активно використовується для відокремлення різноманітного програмного забезпечення, підвищення захищеності системи. Також завдяки роботі з контейнерами можна дуже швидко оновлювати програмне забезпечення (виконувати оновлення на льоту) просто перевикаючи готові контейнери. Також ця система дозволяє безпечно тестувати програми перед їх впровадженням в роботу.

Наразі найвідомішими системами контейнеризації є:

— Docker. Docker це система контейнеризації програм без збереження стану. Його активно використовують у комерції для розгортання серверних програм. В тому числі для реалізації мікросервісної архітектури;
— LXC. LXC це система контейнеризації системи. Його контейнери за замовчуванням зберігають свій стан, але їх також можна налаштувати без збереження стану. Контейнери LXC є більш функціональними й універсальними, але важчими за контейнери Docker.

Docker є сильно автоматизованим засобом, через що впровадження власних методик управління є дещо ускладненим. До того ж через її автоматизованість вона може бути зламаною недоброчесними постачальниками образів або через неуважність адміністратора. Ця система призначена для контейнеризації програм, тобто одного процесу. Через що має обмежений спектр завдань.

Своєю чергою LXC є повністю ручним засобом, надаючи користувачеві лише саму контейнеризацію. Всі інші аспекти контролю контейнерів можна запроваджувати власними засобами. Таким чином ця система є значно надійнішою, хоч і складнішою у використанні. До того ж LXC призначений для контейнеризації операційної системи. Завдяки чому має значно більший спектр задач.

Виходячи з описаних вище властивостей, обираємо систему контейнеризації LXC.

3.2.2. Встановлення та налаштування

Для кафедрального сервера обрано систему контейнеризації LXC. Для її встановлення встановлюємо відповідний пакет.

```
[root@server ~]# pacman -Syyu lxc
```

Для автоматичного запуску та ініціалізації системи вмикаємо відповідну службу. Ця служба також виконує автоматичний запуск контейнерів, що налаштовані для автоматичного запуску.

```
[root@server ~]# systemctl enable --now lxc.service
```

Для запровадження моніторингу станів контейнерів активуємо відповідну допоміжну службу.

```
[root@server ~]# systemctl enable --now lxc-monitord.service
```

Важливо переконатися що в конфігураціях відсутній параметр USE_LXC_BRIDGE="false". Зазвичай його визначають у файлі /etc/default/lxc-net. Це необхідно, щоб заборонити LXC створювати мережевий інтерфейс та налаштувати його. Цим займатиметься systemd-networkd [22].

3.2.3. Налаштування мережі

Створюємо мережевий інтерфейс, за допомогою якого контейнери будуть комунікувати з системою та мати доступ до зовнішньої мережі. За замовчуванням LXC використовує інтерфейс з назвою lxcbr0, тому використовуватимемо цю назву. Завдяки цьому не потрібно буде реконфігурувати контейнери.

Використовуватимемо пристрій типу bridge. Для його створення та конфігурування скористаємось службою systemd-networkd [22]. Для створення інтерфейсу створюємо файл lxcbr0.netdev [30] в каталозі конфігурацій. Вміст цього файлу декларує параметри інтерфейсу, його наведено далі.

```
[NetDev]
Description=Interface for LXC containers
Name=lxcbr0
Kind=bridge
```

Для конфігурування пристрою створюємо файл lxcbr0.network [21] в каталозі конфігурацій. Його вміст наведено нижче.

```
[Match]
Name=lxcbr0

[Network]
Description=Default interface for LXC containers
LinkLocalAddressing=no
Address=10.0.3.1/24
```

Для застосування внесених змін можемо не перезавантажувати сервер, а перезавантажити конфігурацію. Для цього оновлюємо конфігурацію systemd.

```
[root@server ~]# systemctl daemon-reload
```

І перезапускаємо сервіс systemd-networkd.

```
[root@server ~]# systemctl restart systemd-networkd.service
```

Для перевірки застосованих змін переглянемо поточні налаштування мережі.

```
[root@server ~]# ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
 ↪  qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eno1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
 ↪  default qlen 1000
    link/ether ac:1f:6b:01:04:1a brd ff:ff:ff:ff:ff:ff
    altname enp4s0f0
    inet 192.168.0.100/24 brd 192.168.0.255 scope global eno1
       valid_lft forever preferred_lft forever
    inet6 fe80::ae1f:6bff:fe01:41a/64 scope link
       valid_lft forever preferred_lft forever
3: eno2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
 ↪  default qlen 1000
    link/ether ac:1f:6b:01:04:1b brd ff:ff:ff:ff:ff:ff
    altname enp4s0f1
    inet 10.0.0.1/24 brd 10.0.0.255 scope global eno2
       valid_lft forever preferred_lft forever
```

```
    inet6 fe80::ae1f:6bff:fe01:41b/64 scope link
       valid_lft forever preferred_lft forever
4: lxcbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN
↪  group default qlen 1000
    link/ether 7e:85:e5:5f:96:df brd ff:ff:ff:ff:ff:ff
```

Таким чином налаштовано головний мережевий інтерфейс для комунікації контейнерів та супервізора.

3.3. Служба доменних імен

3.3.1. Загальні відомості

Одним з найважливішим елементом комплексної мережі є DNS сервер. Він може працювати в багатьох режимах, надаючи користувачам відповідні можливості. Є 3 основні режими роботи DNS:

— проксі або кеш-сервер. Цей режим використовується для перенаправлення запитів користувачів та реалізації централізованого кешу запитів. Таким чином зменшується навантаження на мережу та на публічні DNS сервери;
— авторитетний. В цьому режимі сервер бере відповідальність за певний домен. Надаючи та контролюючи його. Завдяки цьому домен може мати свій власний простір доменних імен;
— гібридний. В цьому режимі сервер працює в авторитетному режимі. Але при отриманні запиту з невідомою назвою обробляє її відповідно до проксі режиму.

Для кафедри обрано варіант з гібридним режимом роботи. Завдяки чому мережа кафедри отримає наступні послуги:

— централізований кеш DNS;
— власний простір імен;
— пришвидшену обробку запитів;
— зменшення навантаження на мережу університету;
— тощо.

Існує досить велика кількість програмного забезпечення, що реалізує ці послуги. З основних можна виділити наступні:

— dnsmasq. Дуже гарне рішення для невеликих мереж та тих хто не знайомий з DNS записами. Він дозволяє досить просто налаштувати сервер Доменних

імен;

— Berkeley Internet Name Domain (BIND). Цей засіб є стандартом, де-факто, для різноманітних мереж. Він прекрасно підійде для невеликих і корпоративних мереж. Також він підтримує більшість методів взаємодії та стандартів.

Для кафедральної мережі обрано програмне забезпечення BIND. Сервіс налаштовуємо у гібридному режимі. Таким чином при наявності відомостей про запитаний ресурс сервер надаватиме ці відомості у відповідь. При відсутності відомостей сервер здійснюватиме рекурсивний пошук.

3.3.2. Встановлення та налаштування

Для встановлення BIND треба встановити відповідний пакет.

```
[root@server ~]# pacman -Syyu bind
```

Тепер ми маємо створити файли DNS зон, які зчитуватиме сервер і надаватиме відповідь. Сервер має мати відомості про дві зони:

— eom.ust.edu.ua — мережа кафедри;

— lxc — мережа контейнерів.

Для налаштування визначених зон треба створити відповідні файли зон у каталозі /var/named:

— eom.ust.edu.ua.zone. Файл зони eom.ust.edu.ua. Його вміст наведено у додатку Р;

— eom.ust.edu.ua.rev. Файл зворотної зони eom.ust.edu.ua. Його вміст наведено у додатку С;

— lxc.zone. Файл зони lxc. Його вміст наведено у додатку Т;

— lxc.rev. Файл зворотної зони lxc. Його вміст наведено у додатку У.

Таким чином визначено головні статичні відомості про зони, котрими володіє сервер доменних імен.

Також необхідно визначити ряд допоміжних файлів зон:

— named.root. Цей файл містить відомості про кореневі сервери. Його можна взяти з сайту www.internic.net. Його дублікат наведено у додатку Ф;

— localhost.zone. Файл зони localhost. Містить відомості про localhost. Його вміст наведено у додатку Х;

— localhost.rev. Файл зворотної зони localhost для мережі Internet Protocol version 4 (IPv4). Його вміст наведено у додатку Ц;
— localhost.ip6.rev. Файл зворотної зони localhost для мережі Internet Protocol version 6 (IPv6). Його вміст наведено у додатку Ш.

На цьому всі необхідні файли зон створені та готові для використання.

Для запровадження можливості оновлення зон доменних імен необхідно згенерувати ключ для підпису оновлень. Завдяки чому оновлювати зони зможе лише автентифікований користувач. Для цього виконаємо наступну команду.

```
[root@server ~]# tsig-keygen -a hmac-sha256 localhost > /var/named/dns.keys.conf
```

Ця команда згенерує ключ за алгоритмом hmac-sha256 та збереже його у файлі dns.keys.conf.

Тепер настав час налаштувати сам сервер BIND. Його конфігураційний файл named.conf розміщений у директорії /etc. Його вміст наведено у додатку Щ.

Для активації сервісу доменних імен маємо виконати наступну команду.

```
[root@server ~]# systemctl enable --now named.service
```

Відредагуємо файл /etc/resolv.conf. Додаємо в нього налаштований сервер доменних імен.

```
[root@server ~]# echo "nameserver 127.0.0.1" > /etc/resolv.conf
```

3.3.3. Тестування

Для перевірки працездатності, виконаємо запит до сервера за допомогою утиліти dig [31]. Виконуємо декілька прямих запитів.

```
[root@server ~]# dig +short localhost
127.0.0.1
[root@server ~]# dig +short server.eom.ust.edu.ua
10.0.0.1
[root@server ~]# dig +short mail.lxc
10.0.3.1
[root@server ~]# dig +short google.com
172.217.19.110
```

Виконуємо декілька зворотних запитів.

```
[root@server ~]# dig +short -x 127.0.0.1
localhost.
[root@server ~]# dig +short -x 10.0.0.1
ns1.eom.ust.edu.ua.
mail.eom.ust.edu.ua.
eom.ust.edu.ua.
server.eom.ust.edu.ua.
[root@server ~]# dig +short -x 10.0.3.1
lxc.
mail.lxc.
ns1.lxc.
```

З виводів бачимо що сервер працює коректно. Він надає відповіді на прямі та зворотні запити.

3.4. DHCP сервер

3.4.1. Загальні відомості

За визначенням з Wikipedia [32], DHCP — це протокол керування мережею, який використовується в комп'ютерних мережах для автоматичного призначення IP адрес та інших параметрів зв'язку пристроям, підключеним до мережі за допомогою архітектури клієнт-сервер.

Сервіс DHCP це дуже корисний мережевий сервіс. Він забезпечує автоматичну конфігурацію та узгодження параметрів різноманітних хостів в мережі. Також слідкує за розподіленням та часом оренди IP адрес в мережі.

Для систем Linux існує багато реалізацій сервісів для запровадження DHCP у мережі. З найпопулярніших можна виділити:

— systemd-networkd. У складі systemd-networkd присутній вбудований DHCP сервер. Він присутній у всіх системах з systemd, але використовується рідко. Це наймінімальніша реалізація DHCP сервера і запроваджує лише найголовніші можливості. Він спокійно може підійти для невеликої простої мережі, щоб не встановлювати якість додаткові сервіси. До того ж він має дуже простий метод налаштування, котрий виконаний в уніфікованому стилі systemd;

— Internet Systems Consortium (ISC) DHCP. Це повнофункціональний DHCP сервер розроблений організацією ISC. Наразі його можна вважати стандар-

том для мережі будь-якого розміру. Має дуже простий метод конфігурації, котрий має JavaScript Object Notation (JSON) подіну структуру. Наразі ISC не рекомендує його використовувати та припинила його розробку, натомість радять використовувати його наступника;

— Dnsmasq. Це легкий програмний комплекс для мережі. Цей комплекс націлений в першу чергу на домашні та невеликі корпоративні мережі. Він дозволяє швидко налаштувати основні мережеві сервіси. Одним із його компонентів є DHCP сервер. Він має досить різноманітний перелік функцій та можливостей. Його конфігурація є дещо складною та заплутаною. Але на щастя він має дуже детальну сторінку мануала. Також наявний дуже якісний приклад конфігурації з детальними коментарями, в якому представлені різноманітні параметри та методи їх використання.

— Kea. Новий повнофункціональний DHCP сервер від ISC. Він є наступником dhcpd. Наразі він не є дуже популярним, але з кожним днем стає все більш популярним. Kea розроблений для мереж найрізноманітнішої конфігурації та навантаження. Тому він однаково прекрасно підійде для невеликих домашніх мереж і для великих корпоративних мереж. Окрім функцій DHCP він має методи для організації сумісної роботи декількох серверів. Завдяки цьому можна розподілити навантаження між декількома серверами. Для цих цілей Kea має підтримку баз даних MySQL і Postgresql. Але повноцінно використовувати цей функціонал можна лише з платною підпискою. Конфігураційні файли складаються у форматі JSON. Конфігурації мають можливість використання користувацьких полів та значень, тим самим розширюючи можливості конфігурації.

Для кафедральної мережі було обрано DHCP сервер Kea. Цей сервіс встановлюватиметься у складі програмного забезпечення гіпервізору. Він відповідатиме за конфігурацію мережі кафедри та мережі контейнерів.

3.4.2. Встановлення та налаштування

3.4.2.1. Встановлення Kea

Встановлюємо пакет із програмним комплексом Kea.

```
[root@server ~]# pacman -Syyu kea
```

Kea це не просто DHCP сервер, а повноцінний програмний комплекс. Тож кожна його частина конфігурується окремо. Цей програмний комплекс має можливість налаштовуватися декількома способами:

— конфігураційний файл. Налаштування зберігаються у конфігураційному файлі і не змінюються протягом роботи;
— база даних. Налаштування зберігаються у базі даних і можуть оновлюватися протягом роботи. Також завдяки цьому декілька серверів можуть працювати в ідентичних мережах та використовувати однакові параметри. Також присутня можливість відокремлення налаштувань для конкретних груп серверів.

Через неможливість повноцінного використання бази даних та відсутність інших DHCP серверів в мережі прийнято рішення про конфігурування за допомогою конфігураційного файлу.

Оскільки Kea це програмний комплекс то всі його компоненти конфігуруються окремо.

3.4.2.2. Налаштування DDNS сервера

Завдяки власному DNS серверу, налаштованому у розділі 3.3, можемо запровадити автоматичну реєстрацію клієнтів у відповідній DNS зоні. Для запровадження цього сервісу необхідно налаштувати компонент kea-ddns. Його конфігурації розміщені у файлі /etc/kea/kea-dhcp-ddns.conf. Опис його конфігурацій доступний на сайті ISC. Вміст конфігураційного файлу наведено у додатку Ю. У додатку використовується підстанова {СЕКРЕТ}. Ця підстанова відповідає полю секрет ключа localhost у файлі /var/named/dns.keys.conf (див. розділ 3.3).

Цей компонент отримує відомості від DHCP сервера. На підставі цих відомостей генерує Name Change Request (NCR) запит і надсилає його DNS серверу.

Для активацій сервісу необхідно активувати відповідну службу.

```
[root@server ~]# systemctl enable --now kea-dhcp-ddns.service
```

3.4.2.3. Налаштування DHCP для IPv4

Для налаштування DHCP для мережі IPv4 необхідно відредагувати конфігураційний файл /etc/kea/kea-dhcp4.conf. Конфігурації описані на сайті ISC. Його

вміст наведено у додатку Я.

Для запуску цього сервісу необхідно активувати відповідну службу

```
[root@server ~]# systemctl enable --now kea-dhcp4.service
```

3.4.2.4. Налаштування DHCP для IPv6

Тимчасово прийнято рішення про невикористання DHCP для мережі IPv6.

3.5. Брандмауер

Всі сервіси сервера запроваджені та налаштовані. Тепер необхідно відредагувати правила брандмауера. Це необхідно для надання доступу клієнтам до відповідних сервісів. Також це треба для надання клієнтам доступу до зовнішньої мережі, бо сервер також виконує роль шлюзу та маршрутизатора.

Відредагований конфігураційний файл брандмауера сервера наведено у додатку АА.

3.6. Сторонні конфігурації

Змінюємо налаштування клієнта SSH для відповідності поточній конфігурації мережі та наявним можливостям. Редагуємо файл /.ssh/config і редагуємо наявні конфігурації.

```
Match host server.eom.ust.edu.ua exec "knock -d 100 %h 0
↪  {ТАЄМНА_ПОСЛІДОВНІСТЬ_СЕРВЕРА}"
Host server.eom.ust.edu.ua
    Hostname server.eom.ust.edu.ua
    User root
    Port 22

Match host admin-usb.eom.ust.edu.ua exec "knock -d 100 %h 0
↪  {ТАЄМНА_ПОСЛІДОВНІСТЬ_ЗМІННОГО_НОСІЯ}"
Host admin-usb.eom.ust.edu.ua
    Hostname admin-usb.eom.ust.edu.ua
    User root
    Port 22
```

ВИСНОВКИ

В цій роботі розглянуто процес розробки, встановлення та налаштування програмного забезпечення кафедри.

Обрано дистрибутив Manjaro з сімейства Linux. Котрий встановлено на сервер. Налаштовано важливі параметри системи та допоміжних служб операційної системи.

В мережі запроваджено ряд мережевих сервісів і налаштовано сервери:

— NTP;
— DNS;
— DHCP.

Також налаштовано синхронізацію між DHCP та DNS серверами за допомогою DDNS. Завдяки чому кожний користувач мережі при підключенні отримує власну DNS назву.

Налаштовано систему контейнеризації LXC. Ця система є однією з найважливіших відправних точок для запровадження нових сервісів.

Налаштовано дві персональні авторитетні зони DNS:

— eom.ust.edu.ua. Домен мережі кафедри;
— lxc. Домен віртуальної мережі LXC.

ПЕРЕЛІК ПОСИЛАНЬ

1. *Sav_vas Infinity0 E.* Unix history-simple. — 08.07.2008. — URL: https://commons.wikimedia.org/wiki/File:Unix_history-simple.svg (дата зверн. 25.04.2023).
2. *GioComitini*. Linux Distribution Timeline. — 20.01.2023. — URL: https://commons.wikimedia.org/wiki/File:Linux_Distribution_Timeline_Dec._2020.svg (дата зверн. 25.04.2023).
3. *Hundewadt F.* [root tip] [How To] Do a manual Manjaro installation : Manjaro Linux Forum. — 11.01.2023. — URL: https://forum.manjaro.org/t/root-tip-how-to-do-a-manual-manjaro-installation/12507/1 (дата зверн. 24.04.2023) ; Section: Contributions.
4. ip(8) : Arch manual pages. — URL: https://man.archlinux.org/man/ip.8 (дата зверн. 04.05.2023).
5. resolv.conf(5) : Arch manual pages. — URL: https://man.archlinux.org/man/core/man-pages/resolv.conf.5.en (дата зверн. 03.06.2023).
6. ping(8) : Arch manual pages. — URL: https://man.archlinux.org/man/core/iputils/ping.8.en (дата зверн. 05.05.2023).
7. ssh(1) : Arch manual pages. — URL: https://man.archlinux.org/man/ssh.1 (дата зверн. 05.05.2023).
8. ssh_config(5) : Arch manual pages. — URL: https://man.archlinux.org/man/core/openssh/ssh_config.5.en (дата зверн. 05.05.2023).
9. knock(1) : Arch manual pages. — URL: https://man.archlinux.org/man/knock.1 (дата зверн. 05.05.2023).
10. lsblk(8) : Arch manual pages. — URL: https://man.archlinux.org/man/lsblk.8 (дата зверн. 24.04.2023).
11. Filesystem Hierarchy Standard. — URL: https://refspecs.linuxfoundation.org/FHS_3.0/fhs/index.html (дата зверн. 05.05.2023).
12. fdisk(8) : Arch manual pages. — URL: https://man.archlinux.org/man/fdisk.8 (дата зверн. 24.04.2023).

13. pacman - ArchWiki. — 31.03.2023. — URL: https://wiki.archlinux.org/title/Pacman (дата зверн. 24.04.2023).

14. pacman(8) : Arch manual pages. — URL: https://man.archlinux.org/man/core/pacman/pacman.8.en (дата зверн. 24.04.2023).

15. mdadm(8) : Arch manual pages. — URL: https://man.archlinux.org/man/mdadm.8 (дата зверн. 07.06.2023).

16. mkinitcpio(8) : Arch manual pages. — URL: https://man.archlinux.org/man/core/mkinitcpio/mkinitcpio.8.en (дата зверн. 23.04.2023).

17. mkinitcpio.conf(5) : Arch manual pages. — URL: https://man.archlinux.org/man/core/mkinitcpio/mkinitcpio.conf.5.en (дата зверн. 23.04.2023).

18. GNU GRUB - GNU Project - Free Software Foundation (FSF). — URL: https://www.gnu.org/software/grub/ (дата зверн. 28.04.2023).

19. fstab(5) : Arch manual pages. — URL: https://man.archlinux.org/man/core/util-linux/fstab.5.en (дата зверн. 28.04.2023).

20. grub-install(8) : Arch manual pages. — URL: https://man.archlinux.org/man/grub-install.8 (дата зверн. 28.04.2023).

21. systemd.network(5) : Arch manual pages. — URL: https://man.archlinux.org/man/systemd.network.5.en (дата зверн. 23.04.2023).

22. systemd-networkd(8) : Arch manual pages. — URL: https://man.archlinux.org/man/systemd-networkd.8 (дата зверн. 23.04.2023).

23. netfilter/iptables project homepage : The netfilter.org project. — URL: https://www.netfilter.org/ (дата зверн. 02.05.2023).

24. nft(8) : Arch manual pages. — URL: https://man.archlinux.org/man/nft.8 (дата зверн. 02.05.2023).

25. Port knocking // Вікіпедія. — 06.02.2023. — URL: https://en.wikipedia.org/w/index.php?title=Port_knocking&oldid=1137766340 (дата зверн. 02.05.2023) ; Page Version ID: 1137766340.

26. passwd(1) : Arch manual pages. — URL: https://man.archlinux.org/man/core/shadow/passwd.1.en (дата зверн. 03.05.2023).

27. sshd(8) : Arch manual pages. — URL: https://man.archlinux.org/man/sshd.8 (дата зверн. 24.04.2023).

28. NTP // Вікіпедія. — 28.07.2022. — URL: https://uk.wikipedia.org/w/index.php?title=NTP&oldid=36678861 (дата зверн. 21.05.2023) ; Page Version ID: 36678861.

29. Віртуалізація на рівні операційної системи // Вікіпедія. — 07.06.2022. — URL: https://uk.wikipedia.org/w/index.php?title=%D0%92%D1%96%D1%80%D1%82%D1%83%D0%B0%D0%BB%D1%96%D0%B7%D0%B0%D1%86%D1%96%D1%8F_%D0%BD%D0%B0_%D1%80%D1%96%D0%B2%D0%BD%D1%96_%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D1%96%D0%B9%D0%BD%D0%BE%D1%97_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B8&oldid=36092331 (дата зверн. 20.05.2023) ; Page Version ID: 36092331.

30. systemd.netdev(5) : Arch manual pages. — URL: https://man.archlinux.org/man/core/systemd/systemd.netdev.5.en (дата зверн. 11.05.2023).

31. dig(1) : Arch manual pages. — URL: https://man.archlinux.org/man/extra/bind/dig.1.en (дата зверн. 20.05.2023).

32. Dynamic Host Configuration Protocol // Вікіпедія. — 01.05.2023. — URL: https://en.wikipedia.org/w/index.php?title=Dynamic_Host_Configuration_Protocol&oldid=1152615284 (дата зверн. 05.05.2023) ; Page Version ID: 1152615284.

33. *Apacer Technology Inc*. Apacer : Apacer. — URL: https://www.facebook.com/Apacer.global (дата зверн. 24.04.2023).

34. lsusb(8) : Arch manual pages. — URL: https://man.archlinux.org/man/core/usbutils/lsusb.8.en (дата зверн. 24.04.2023).

35. mkfs(8) : Arch manual pages. — URL: https://man.archlinux.org/man/mkfs.8 (дата зверн. 24.04.2023).

36. e2label(8) : Arch manual pages. — URL: https://man.archlinux.org/man/core/e2fsprogs/e2label.8.en (дата зверн. 24.04.2023).

37. blkid(8) : Arch manual pages. — URL: https://man.archlinux.org/man/blkid.8 (дата зверн. 24.04.2023).

38. sort(1) : Arch manual pages. — URL: https://man.archlinux.org/man/sort.1 (дата зверн. 24.04.2023).

39. systemd.net-naming-scheme(7) : Arch manual pages. — URL: https://man.archlinux.org/man/systemd.net-naming-scheme.7 (дата зверн. 23.04.2023).

40. rsync(1) : Arch manual pages. — URL: https://man.archlinux.org/man/rsync.1 (дата зверн. 02.05.2023).

41. mount(8) : Arch manual pages. — URL: https://man.archlinux.org/man/mount.8 (дата зверн. 22.05.2023).

42. cp(1) : Arch manual pages. — URL: https://man.archlinux.org/man/core/coreutils/cp.1.en (дата зверн. 28.04.2023).

43. ls(1) : Arch manual pages. — URL: https://man.archlinux.org/man/ls.1 (дата зверн. 24.04.2023).

44. umount(8) : Arch manual pages. — URL: https://man.archlinux.org/man/umount.8 (дата зверн. 28.04.2023).

45. rmdir(1) : Arch manual pages. — URL: https://man.archlinux.org/man/rmdir.1 (дата зверн. 28.04.2023).

46. hostname(5) : Arch manual pages. — URL: https://man.archlinux.org/man/core/systemd/hostname.5.en (дата зверн. 24.04.2023).

47. hosts(5) : Arch manual pages. — URL: https://man.archlinux.org/man/hosts.5 (дата зверн. 24.04.2023).

48. grep(1) : Arch manual pages. — URL: https://man.archlinux.org/man/grep.1 (дата зверн. 24.04.2023).

49. vconsole.conf(5) : Arch manual pages. — URL: https://man.archlinux.org / man / core / systemd / vconsole . conf . 5 . en (дата зверн. 23.04.2023).

50. locale.gen(5) : Debian Manpages. — URL: https://manpages.debian.org / bullseye / locales / locale . gen . 5 . en . html (дата зверн. 24.04.2023).

51. locale-gen(8) : Debian Manpages. — URL: https://manpages.debian.org/bullseye/locales/locale-gen.8.en.html (дата зверн. 24.04.2023).

52. locale.conf(5) : Arch manual pages. — URL: https://man.archlinux.org/man/core/systemd/locale.conf.5.en (дата зверн. 24.04.2023).

53. localtime(5) : Arch manual pages. — URL: https://man.archlinux.org/man/core/systemd/localtime.5.en (дата зверн. 24.04.2023).

54. ln(1) : Arch manual pages. — URL: https://man.archlinux.org/man/core/coreutils/ln.1.en (дата зверн. 24.04.2023).

55. sshd_config(5) : Arch manual pages. — 03.03.2023. — URL: https://man.archlinux.org/man/core/openssh/sshd_config.5.en (дата зверн. 23.04.2023).

56. *Pornin T.* Answer to "SSH key strength" : Information Security Stack Exchange. — 21.02.2016. — URL: https://security.stackexchange.com/a/115296 (дата зверн. 24.04.2023).

57. *Katz A.* Answer to "What are ssh-keygen best practices?" : Information Security Stack Exchange. — 30.11.2016. — URL: https://security.stackexchange.com/a/144044 (дата зверн. 24.04.2023).

58. *Katz A.* Answer to "What's the difference between id_rsa.pub and id_dsa.pub?" : Stack Overflow. — 09.01.2015. — URL: https://stackoverflow.com/a/27855045 (дата зверн. 24.04.2023).

59. ssh-keygen(1) : Arch manual pages. — 10.02.2023. — URL: https://man.archlinux.org/man/ssh-keygen.1.en (дата зверн. 23.04.2023).

60. mkdir(1) : Arch manual pages. — URL: https://man.archlinux.org/man/core/coreutils/mkdir.1.en (дата зверн. 24.04.2023).

61. chmod(1) : Arch manual pages. — URL: https://man.archlinux.org/man/chmod.1 (дата зверн. 24.04.2023).

ДОДАТОК А
ОПИС СІМЕЙСТВА ОПЕРАЦІЙНИХ СИСТЕМ WINDOWS

Сімейство операційних систем Windows розробляється компанією Microsoft і бере свій початок від операційної системи Disk Operating System (DOS). Розповсюджується централізовано. Весь первинний код цього сімейства є закритим, за винятком певних компонентів.

Операційні системи сімейства Windows дуже популярні та активно використовується на користувацьких комп'ютерах. Як серверні системи, це сімейство використовується лише для адміністративних серверів, у мережах Windows, завдяки наявності відповідного програмного забезпечення. Дуже рідко вони використовуються для робочих серверів з середнім та високим навантаженням.

Головною перевагою цього сімейства є простота для користувача, хоча цього не можна сказати про останні системи сімейства.

Windows мають великі проблеми з надійністю, швидкодією та захищеністю. Системи сімейства Windows є дуже вразливими до вторгнення та шкідливого програмного забезпечення і потребують встановлення великої кількості додаткових систем захисту. До того ж оновлення системи відбуваються досить рідко, тому відомі вразливості залишаються активними протягом досить великого проміжку часу. Відсутність системи автоматичного оновлення користувацького забезпечення призводить до використання користувачами застарілих версій. Всі перелічені фактори призводять до значного зменшення швидкодії, надійності, та захищеності операційної системи цього сімейства, а також до проблем сумісності та підтримки сучасних протоколів. Їх використання є дуже поганою практикою в корпоративній сфері, але їх все ж таки активно там використовують, "бо звично".

Агресивний маркетинг Microsoft зробив з деяких їх програм "стандарти наприклад Microsoft Office. Але навіть Microsoft Office має величезні проблеми сумісності версій, причиною цього є відсутність стандарту його форматів. Багато хто зустрічав проблему коли документ з однієї версії Office некоректно читається в іншій. Тому їх використання не рекомендується.

ДОДАТОК Б
ОПИС СІМЕЙСТВА ОПЕРАЦІЙНИХ СИСТЕМ LINUX

Сімейство Linux здебільшого розробляється різноманітними спільнотами розробників, але деякі операційні системи розробляються конкретними компаніями або при їх підтримці, наприклад RedHat, Google, Canonical, Intel, Steam та інші.

Хоч операційні системи цього сімейства зазвичай мають багато відмінностей, але ядро, модулі ядра, технології, принципи та програмне забезпечення, зазвичай однакові. Тому їх заведено називати дистрибутивами Linux. Тому для коректності, окремі представники сімейства Linux будуть так називатися.

Важливо розуміти відмінність між різними поняттями слова дистрибутив в залежності від контексту.

У загальному значенні, дистрибутив — це форма розповсюдження програмного забезпечення.

У контексті програмного забезпечення, дистрибутив — це образ програми, що використовується для його розповсюдження. Цей образ може мати багато виглядів:

— образ файлової системи;
— архів;
— програма-встановлювач;
— пакет.

У різних операційних системах використовуються різні види дистрибутивів програмного забезпечення:

— Windows зазвичай використовує архіви (для портативного програмного забезпечення), EXE-встановлювачі або MSI-встановлювачі;
— Mac OS переважно використовує образи DMG;
— Linux використовує пакети (RPM, DEB тощо), архіви зі скомпільованою програмою (так звані tarball-и)), а також у вигляді початкового коду, що заархівований в форматі tar.gz або tar.bz2.

У контексті операційних систем, дистрибутив — це готовий образ системи, що використовується для встановлення на інші комп’ютери. До його можливостей входить ініціалізація апаратної частини, завантаження урізаної версії системи та запуск програми-встановлювача.

Але у випадку з дистрибутивами Linux термін дистрибутив використовується не тільки для встановлювачів, а й для встановленої за його допомогою системи. Цей термін може використовуватися у скороченій формі — дистро.

До дистрибутивів Linux відносяться операційні системи, що складаються зі системних бібліотек та інструментів, розроблених проєктом GNU, ядра Linux та інших програм.

Весь первинний код операційних систем сімейства Linux відкритий, за винятком певних компонентів конкретних дистрибутивів, наприклад: Android, ChromOS, RHEL. Цей факт дозволяє всім охочим ознайомлюватися та аналізувати первинний код на наявність шкідливого програмного забезпечення, помилок, вразливостей. Завдяки цьому всі проблеми дуже оперативно виявляються та виправляються. Завдяки чому системи сімейства Linux дуже надійні та відмовостійкі.

Навіть базові налаштування системи безпеки забезпечують високий рівень захисту від вторгнення та шкідливих програм. В разі використання правильно налаштованої мандатної системи безпеки, AppArmor або SELinux, робить це сімейство неймовірно захищеним від будь-якого вторгнення та шкідливого програмного забезпечення.

Майже всі дистрибутиви Linux мають системи автоматичної дистрибуції, оновлення та узгодження програмного забезпечення. Тому всі компоненти системи та користувацькі програми підтримуються в актуалізованому стані. Завдяки цьому досить оперативно виправляються всі виявлені помилки та вразливості безпеки в програмах, що дуже позитивно впливає на відмовостійкість та захищеність системи.

Завдяки різноманітності сімейства, а особливо його програм, було сформовано дуже багато стандартів та домовленостей і кожний розробник слідує цим документам. Завдяки чому всі програми мають повну сумісність з більшістю операційних систем сімейства.

Як приклади стандартів можна навести:

— Portable Operating System Interfaces for unix (POSIX) або ISO/IEC 9945;
— Single UNIX Specification (SUS);
— Linux Standard Base (LSB) або ISO/IEC 23360;
— Open Document Format (ODF), або Open Document Format for Office Application, або ISO/IEC 26300;

— FHS [11];
— FreeDesktop;
— тощо.

Завдяки POSIX та SUS забезпечується сумісність програми з системою. Завдяки FHS [11] програми знають де шукати та зберігати необхідні дані. Завдяки FreeDesktop програми отримують сумісність з різними графічними оболонками. Завдяки ODF досягається повна сумісність документів створених у різних офісних пакетах та/чи редакторах.

Також важливо зауважити, хоча повна сумісність є правилом, воно має свої винятки та досить багато, але вони є доцільно обґрунтованими. Прикладом такого винятку є неповна сумісність "класичних"дистрибутивів та дистрибутивів сімейства Android. Їх неповна сумісність зумовлена використанням значно легшої та простішої системної бібліотеки Bionic в Android замість glibc, що використовується у "класичних"дистрибутивах.

До того ж оптимізація, швидкодія та гнучкість цих систем робить їх ідеальними кандидатами для серверів з високим навантаженням. А наявність величезної кількості мережевих компонентів та їх сумісність робить це сімейство ідеальними кандидатами для будь-якої мережевої системи.

Крім того, величезна різноманітність програмного забезпечення значно ускладнює пошук та експлуатацію вразливостей. Кожна система має свою унікальну конфігурацію і, відповідно, свій власний унікальний набір вразливостей. Що значно підвищує надійність сімейства. А періодична заміна застарілого забезпечення на нове значною мірою підвищує надійність конкретного дистрибутиву. Тобто вразливості однієї системи не працюватимуть в іншій системі або версії.

Всі описані фактори роблять дистрибутиви Linux ідеальними кандидатами для використання в будь-якій сфері.

ДОДАТОК В
СТВОРЕННЯ НОСІЯ АДМІНІСТРАТОРА

В.1. Вибір носія

Обираємо змінний носій. На цей носій встановиться портативна система. Ця система виконуватиме дуже важливу роботу, допомагаючи в адмініструванні мережі, сервера та клієнтських комп'ютерів.

Для цих цілей можна використовувати live версію системи, що надається розробниками дистрибутивів, але вони не зберігають зміни під час перезавантаження системи. Тому необхідно виконати повну інсталяцію системи на змінний носій.

Ще одним важливим аспектом є необхідність підтримки завантаження з цього носія у двох режимах: BIOS та UEFI. Завдяки чому носій стане сумісним зі старими та новими комп'ютерами. Що значною мірою полегшить використання носія, розширить його спектр можливостей і позбавить необхідності в другому носії для іншої системи.

Обрано змінний USB носій Apacer AH333. Його характеристики з офіційного сайту виробника [33] наведені у таблиці В.1.

Таблиця В.1 — Специфікація Flash-носія Apacer AH333

Характеристика	Значення
Модель	AH333 USB 2.0 Flash Drive
Колір	Таємничий чорний
Об'єм	32 ГБ
Інтерфейс	Високошвидкісний USB 2.0, сертифікований зворотно сумісний з USB 1.1
Джерело живлення	Живлення від шини USB; Зовнішнє живлення не потрібне
Сумісний зі системами	Windows 10/8.1/8/7,Mac OS 10.6.X або вище, Linux 2.6.X або вище Робоча температура 0°C до +60°C (32°F до 140°F)

Цей носій повною мірою задовольняє всім потребам, навіть з запасом, і має повну сумісність з більшістю комп'ютерів.

В.2. Ідентифікація пристрою

Під'єднуємо пристрій до комп'ютера і перевіряємо його справність. Для перевірки цього виконаємо команду lsusb [34].

```
[root@hostname ~]# lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 002: ID 1bcf:2b98 Sunplus Innovation Technology Inc.
↪  Integrated_Webcam_HD
Bus 001 Device 004: ID 1005:b113 Apacer Technology, Inc. Handy Steno/AH123 / Handy
↪  Steno 2.0/HT203
Bus 001 Device 003: ID 0cf3:e009 Qualcomm Atheros Communications
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

У переліку ідентифікованих пристроїв бачимо встановлений носій, а отже система його бачить і він стравний.

Далі ідентифікуємо файл, з яким система пов'язала носій. Для цього виконуємо команду lsblk [10], з прапорами noempty (не виводити порожні пристрої) та scsi (вивести дані щодо пристроїв Small Computer Systems Interface (SCSI)).

```
[root@hostname ~]# lsblk --noempty --scsi
NAME HCTL       TYPE VENDOR   MODEL                     REV SERIAL               TRAN
sda  0:0:0:0    disk ATA      ST1000LM035-1RK172     SDM3 WL1PSGL4               sata
sdb  2:0:0:0    disk           USB FLASH DRIVE        PMAP 07010B32CF82E347     usb
sr0  1:0:0:0    rom  PLDS     PLDS DVD+/-RW DU-8A5LH 6D1M PNDVVPLC0089M5B9OA00 sata
```

З виводу бачимо, що система пов'язала дисковий пристрій з файлом sdb. Знаючи назву файлу пристрою можемо приступити до його налаштування.

В.3. Форматування носія

В.3.1. Опис

Носій має дещо специфічні налаштування дискового простору у порівнянні з класичними.

Розділ 1 використовується для резервування місця в початку дискового простору. Цей розділ є необхідним для диска з таблицею розділів GUID Partition Table (GPT), щоб забезпечити можливість завантаження системи в режимі BIOS. BIOS шукає та зчитує завантажувач розміщений у перших секторах дискового простору. А через відсутність резервного простору в таблиці GPT завантажувач не зможе

встановитися, або викличе збій. Для цього необхідно зарезервувати простір під завантажувач. В специфікаціях та рекомендаціях вказано розмір 8 MiB. Він повинен мати тип BIOS boot. Жодної файлової системи він не потребує. Адже цей простір буде перезаписаний завантажувачем.

Розділ 2 призначений для розміщення завантажувача, що буде використовуватися при завантаженні в режимі UEFI. Специфікація щодо його розміру відсутня. Тому створюємо розділ з запасом - 512 MiB. Він повинен мати тип EFI System. Також потребує файлову систему FAT32, бо вона входить у стандарт UEFI.

Розділ 3 призначений для самої системи. В ньому знаходитимуться всі файли системи. Тому він займатиме весь доступний простір. Використаємо стандартну файлову систему Linux — Ext4. Це оптимальна за показниками файлова система і призначена для широкого спектра задач.

Структурований опис параметрів наведено у таблиці В.2

Таблиця В.2 — Перелік необхідних розділів та їх параметрів

Номер	Об’єм	Тип	Файлова система
1	8 MiB	BIOS boot	—
2	512 MiB	EFI System	FAT32
3	Весь доступний простір	Linux filesystem	Ext4

В.3.2. Розмітка дискового простору

Для внесення відповідних змін (див. табл. В.2) використаємо утиліту fdisk [12]. Інструкція з її використання наведена у додатку Г

Для перевірки застосованих дій виконуємо команду lsblk [10], але без ключа scsi.

```
[root@hostname ~]# lsblk --noempty
NAME        MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
sda           8:0    0 931,5G  0 disk
├─sda1        8:1    0    32G  0 part [SWAP]
├─sda2        8:2    0   200G  0 part /var
├─sda3        8:3    0    50G  0 part /tmp
└─sda4        8:4    0 649,5G  0 part /home
sdb           8:16   1  28,9G  0 disk
├─sdb1        8:17   1     8M  0 part
├─sdb2        8:18   1   512M  0 part
└─sdb3        8:19   1  28,4G  0 part
```

```
sr0          11:0    1  1024M  0 rom
nvme0n1     259:0    0 465,8G  0 disk
├─nvme0n1p1 259:1    0     8M  0 part
├─nvme0n1p2 259:2    0     1G  0 part /boot/efi
└─nvme0n1p3 259:3    0 464,8G  0 part /
```

Повною мірою переконавшись в коректності внесених змін, продовжуємо подальше налаштування.

В.4. Ініціалізація файлових систем

Далі ініціалізуємо файлові системи у розділах диска. Необхідні файлові системи для розділів системи наведені у таблиці В.2.

Для їх створення та ініціалізації скористаємось утилітою mkfs [35]. Інструкція з ініціалізації файлових систем наведена у додатку Д

Також, для зручності в ідентифікації пристрою та файлової системи у майбутньому, додамо файловій системі, у розділі 3, ярлик. Це можна зробити за допомогою команди e2label [36]. Цій програмі необхідно вказати шлях до розділу sdb3 і назву ярлика. Враховуючи призначення та носій розміщення, дамо йому назву Admin USB. В результаті отримуємо команду.

```
[root@hostname ~]# e2label /dev/sdb3 "Admin USB"
```

Таким чином портативний носій адміністратора повністю налаштований.

Наостанок необхідно виписати ідентифікатори файлових систем. Для цього виконаємо команду blkid [37] і відсортуємо вивід за допомогою програми sort [38].

```
[root@hostname ~]# blkid | sort
/dev/nvme0n1p1: PARTUUID="03f70e97-fb25-654b-acae-b2dd2036a894"
/dev/nvme0n1p2: UUID="1A65-5145" BLOCK_SIZE="512" TYPE="vfat"
↪  PARTUUID="21a0d8ee-476f-6249-aebf-6f4e4a1db62c"
/dev/nvme0n1p3: UUID="2bfeda90-1d1a-4702-9103-6f9dffa59979" BLOCK_SIZE="4096"
↪  TYPE="ext4" PARTUUID="f180b020-186f-a745-a71f-d6bf0d3d142e"
/dev/sda1: UUID="0992f05f-240b-44f2-bcd0-78ab285dd28a" TYPE="swap"
↪  PARTUUID="0adb2e10-07dd-8947-b16e-d5a06eb66019"
/dev/sda2: UUID="44c5e289-d850-4632-bb6f-818810a0a691" BLOCK_SIZE="4096"
↪  TYPE="ext4" PARTUUID="5ae695ba-23d1-6343-85dc-1dd106b909a0"
/dev/sda3: UUID="ef28525a-d46b-46df-b055-da4a608b4f37" BLOCK_SIZE="4096"
↪  TYPE="ext4" PARTUUID="23249808-cee1-5741-b784-fa47fa82fd16"
/dev/sda4: UUID="30dbf37f-51dc-4b67-b70e-e4d47a22fe98" BLOCK_SIZE="4096"
↪  TYPE="ext4" PARTUUID="6c841c13-2f51-2c4a-87ba-fa1748346011"
```

```
/dev/sdb1: PARTUUID="f0ff44c3-d287-fa49-b773-f760bd1cd890"
/dev/sdb2: UUID="3698-E89C" BLOCK_SIZE="512" TYPE="vfat"
↪  PARTUUID="df636719-b877-f04f-a336-97cf5ec162bc"
/dev/sdb3: UUID="149f0424-e581-4b4f-9002-fa8fbe09e55b" BLOCK_SIZE="4096"
↪  TYPE="ext4" PARTUUID="4173c474-8857-5246-b560-096b07a23995"
```

На цьому налаштування змінного носія завершено. Сформуємо кінцеву інформацію про конфігурацію носія. Вона наведена у таблиці В.3.

Таблиця В.3 — Кінцева конфігурація носія

Номер	Ідентифікатор	Ярлик	Файлова система	Точка монтування
1	—	—	—	—
2	3698-E89C	—	FAT32	/efi
3	149f0424-e581-4b4f-9002-fa8fbe09e55b	Admin USB	Ext4	/

В.5. Збірка операційної системи

В.5.1. Підготовка віртуального середовища

Після налаштування носія, треба інсталювати операційну систему. Враховуючи повільну роботу USB носіїв, збірку операційної системи виконуємо на комп’ютері. Згодом готову систему скопіюємо на носій. Завдяки цьому зберігаємо багато часу на очікувані зчитування та запису.

Процес збірки та встановлення системи базується на записі розробників дистрибутиву з форуму Manjaro [3].

Створюємо директорію, в якій виконуватимемо збірку та конфігурування систему.

```
[root@hostname ~]# mkdir USB_OS
```

Виконуємо ініціалізацію базового образу системи за допомогою basestrap. Вкажемо їх шлях до директорії та пакунок base. Вказувати додаткові пакети для

встановлення на цьому етапі не бажано. При вказуванні інших пакетів для встановлення може відбутися помилка через відсутність інтерактивного режиму. Тому всі інші пакети встановлюватимуться пізніше.

```
[root@hostname ~]# basestrap /home/bogdan/USB_OS/ base
```

Виконаємо chroot-тизацію в цю директорію з використанням утиліти manjaro-chroot. Таким чином створюється ізольоване віртуальне середовище, в якому значно легше та безпечніше працювати.

```
[root@hostname ~]# manjaro-chroot /home/bogdan/USB_OS/ /bin/bash
```

Оновлюємо бази даних пакетів пакетного менеджера pacman [13; 14].

```
[root@hostname /]# pacman -Syyu
```

В.5.2. Встановлення мінімально необхідного набору пакетів

Встановлюємо необхідні для роботи системи пакети:

- linux61. Ядро Linux 6.1;
- linux-firmware. Набір мікропрограм низького рівня (драйверів);
- grub. Завантажувач операційної системи;
- grub-theme-manjaro. Тема для завантажувача від розробників дистрибутиву (опціонально);
- efibootmgr. Програма для адміністрування UEFI Boot Manager;
- mkinitcpio. Утиліта для генерації initramfs образів;
- mdadm. Система для взаємодії з RAID масивами;
- manjaro-release. Містить інформацію про поточний реліз системи та надає інформацію про нього іншим програмам при запиті;
- nano. Текстовий редактор.

Додамо підтримку RAID масивів. Деякі контролери RAID повністю абстрагують систему від керування. Інші ж цього не роблять. Тому дуже бажано встановити та налаштувати сервіс mdadm [15] під час завантаження, про всяк випадок. Завдяки цьому носій зможе налаштовувати та взаємодіяти з RAID масивами.

Для встановлення перелічених вище пакетів виконуємо наступну команду.

```
[root@hostname /]# pacman -Syyu linux61 linux-firmware grub grub-theme-manjaro
↪  efibootmgr mkinitcpio mdadm manjaro-release nano
```

В.5.3. Налаштування системи завантаження

В.5.3.1. Налаштування системи збірки

Для генерації островів використовуватимемо систему збірки mkinitcpio [16]. Її налаштування відбувається шляхом редагування файл конфігурації. Цей файл розміщений за шляхом /etc/mkinitcpio.conf [17].

Для забезпечення підтримки RAID, додаємо гачок mdadm_udev після гачка udev.

Вмикаємо стиснення островів алгоритмом gzip, призначивши параметру COMPRESSION значення gzip. І вимикаємо розархівування модулів ядра, надавши параметру MODULES_DECOMPRESS значення no.

Повний файл конфігурації наведено нижче.

```
# vim:set ft=sh
# MODULES
# The following modules are loaded before any boot hooks are
# run.  Advanced users may wish to specify all system modules
# in this array.  For instance:
#     MODULES=(usbhid xhci_hcd)
MODULES=()

# BINARIES
# This setting includes any additional binaries a given user may
# wish into the CPIO image.  This is run last, so it may be used to
# override the actual binaries included by a given hook
# BINARIES are dependency parsed, so you may safely ignore libraries
BINARIES=()

# FILES
# This setting is similar to BINARIES above, however, files are added
# as-is and are not parsed in any way.  This is useful for config files.
FILES=()

# HOOKS
# This is the most important setting in this file.  The HOOKS control the
# modules and scripts added to the image, and what happens at boot time.
# Order is important, and it is recommended that you do not change the
# order in which HOOKS are added.  Run 'mkinitcpio -H <hook name>' for
# help on a given hook.
# 'base' is _required_ unless you know precisely what you are doing.
# 'udev' is _required_ in order to automatically load modules
# 'filesystems' is _required_ unless you specify your fs modules in MODULES
# Examples:
```

```
##   This setup specifies all modules in the MODULES setting above.
##   No RAID, lvm2, or encrypted root is needed.
#    HOOKS=(base)
#
##   This setup will autodetect all modules for your system and should
##   work as a sane default
#    HOOKS=(base udev autodetect modconf block filesystems fsck)
#
##   This setup will generate a 'full' image which supports most systems.
##   No autodetection is done.
#    HOOKS=(base udev modconf block filesystems fsck)
#
##   This setup assembles a mdadm array with an encrypted root file system.
##   Note: See 'mkinitcpio -H mdadm_udev' for more information on RAID devices.
#    HOOKS=(base udev modconf keyboard keymap consolefont block mdadm_udev encrypt
 ↪  filesystems fsck)
#
##   This setup loads an lvm2 volume group.
#    HOOKS=(base udev modconf block lvm2 filesystems fsck)
#
##   NOTE: If you have /usr on a separate partition, you MUST include the
#    usr and fsck hooks.
HOOKS=(base udev autodetect modconf kms keyboard keymap consolefont block
 ↪  mdadm_udev filesystems fsck)

# COMPRESSION
# Use this to compress the initramfs image. By default, gzip compression
# is used. Use 'cat' to create an uncompressed image.
COMPRESSION="gzip"
#COMPRESSION="bzip2"
#COMPRESSION="lzma"
#COMPRESSION="xz"
#COMPRESSION="lzop"
#COMPRESSION="lz4"
#COMPRESSION="zstd"

# COMPRESSION_OPTIONS
# Additional options for the compressor
#COMPRESSION_OPTIONS=()

# MODULES_DECOMPRESS
# Decompress kernel modules during initramfs creation.
# Enable to speedup boot process, disable to save RAM
# during early userspace. Switch (yes/no).
MODULES_DECOMPRESS="no"
```

В.5.3.2. Налаштування завантажувача

Для завантаження системи використовуватимемо завантажувач GRUB [18]. Для його налаштування відредагуємо файл /etc/default/grub.

За замовчуванням завантажувач одразу намагається завантажити систему. Тому для надання можливості змінити його поведінку при завантаженні додаємо затримку завантаження довжиною 5 секунд. Як поведінку затримки обрано варіант menu.

```
GRUB_TIMEOUT=5
GRUB_TIMEOUT_STYLE=menu
```

Для запобігання плутанини та генерації непотрібних пунктів меню завантажувача вимкнено автоматичний пошук встановлених систем.

```
GRUB_DISABLE_OS_PROBER=true
```

Повний відредагований файл конфігурацій наведено нижче.

```
# GRUB boot loader configuration

GRUB_DEFAULT=saved
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR="Manjaro"
GRUB_CMDLINE_LINUX_DEFAULT="quiet udev.log_priority=3"
GRUB_CMDLINE_LINUX=""

# Preload both GPT and MBR modules so that they are not missed
GRUB_PRELOAD_MODULES="part_gpt part_msdos"

# Uncomment to enable booting from LUKS encrypted devices
#GRUB_ENABLE_CRYPTODISK=y

# Set to 'countdown' or 'menu' to change timeout behavior,
# press ESC key to display menu.
GRUB_TIMEOUT_STYLE=menu

# Uncomment to use basic console
GRUB_TERMINAL_INPUT=console

# Uncomment to disable graphical terminal
#GRUB_TERMINAL_OUTPUT=console
```

```
# The resolution used on graphical terminal
# note that you can use only modes which your graphic card supports via VBE
# you can see them in real GRUB with the command 'videoinfo'
GRUB_GFXMODE=auto

# Uncomment to allow the kernel use the same resolution used by grub
GRUB_GFXPAYLOAD_LINUX=keep

# Uncomment if you want GRUB to pass to the Linux kernel the old parameter
# format "root=/dev/xxx" instead of "root=/dev/disk/by-uuid/xxx"
#GRUB_DISABLE_LINUX_UUID=true

# Uncomment to disable generation of recovery mode menu entries
GRUB_DISABLE_RECOVERY=true

# Uncomment and set to the desired menu colors.  Used by normal and wallpaper
# modes only.  Entries specified as foreground/background.
GRUB_COLOR_NORMAL="light-gray/black"
GRUB_COLOR_HIGHLIGHT="green/black"

# Uncomment one of them for the gfx desired, a image background or a gfxtheme
#GRUB_BACKGROUND="/usr/share/grub/background.png"
GRUB_THEME="/usr/share/grub/themes/manjaro/theme.txt"

# Uncomment to get a beep at GRUB start
#GRUB_INIT_TUNE="480 440 1"

# Uncomment to make GRUB remember the last selection. This requires
# setting 'GRUB_DEFAULT=saved' above.
GRUB_SAVEDEFAULT=true

# Uncomment to disable submenus in boot menu
#GRUB_DISABLE_SUBMENU=y

# Uncomment this option to enable os-prober execution in the grub-mkconfig command
GRUB_DISABLE_OS_PROBER=true

# Uncomment to ensure that the root filesystem is mounted read-only so that
# systemd-fsck can run the check automatically. We use 'fsck' by default, which
# needs 'rw' as boot parameter, to avoid delay in boot-time. 'fsck' needs to be
# removed from 'mkinitcpio.conf' to make 'systemd-fsck' work.
# See also Arch-Wiki: https://wiki.archlinux.org/index.php/Fsck#Boot_time_checking
#GRUB_ROOT_FS_RO=true
```

В.5.3.3. Налаштування автоматичного монтування файлових систем

Файл /etc/fstab [19] це конфігураційний файл системи де містяться відомості про пристрої та їх точки монтування, котрі необхідно змонтувати при ініціалізації системи. Це дуже важливий файл і вимагає додаткової уваги. Адже невірна конфігурація чи помилка в одній літері може призвести до неможливості завантаження системи.

Його синтаксис побудований у вигляді таблиці з шістьма колонками:

1. ідентифікатор;
2. точка монтування;
3. файлова система;
4. параметри монтування;
5. необхідність dump;
6. пріоритет перевірки файлової системи.

Використовуючи дані про форматування дискового простору з таблиці В.3, створюємо відповідні записи у файлі /etc/fstab [19].

Готовий вміст файлу /etc/fstab [19] наведено нижче.

```
# Static information about the filesystems.
# See fstab(5) for details.

# <file system> <dir> <type> <options> <dump> <pass>
UUID=149f0424-e581-4b4f-9002-fa8fbe09e55b       /       ext4    defaults
↪  0       1
UUID=3698-E89C                                  /efi    vfat
↪  defaults,fat=32,umask=0077      0       2
tmpfs                                           /tmp    tmpfs
↪  defaults,size=2G,mode=1777      0       0
```

В.5.4. Налаштування мережі

Носій адміністратора це допоміжний інструмент, який дозволить ініціалізувати або відновити систему. Тому він має оброблятися мережею як звичайний користувацький комп’ютер. Тобто сервер має видавати йому IP адресу та конфігурувати його. Для цього необхідно встановити та активувати DHCP клієнт. В якості DHCP клієнта обрано вбудовану в systemd службу DHCP.

Для налаштування цієї служби необхідно створити файл .network [21] в

директорії /etc/systemd/network. Назвемо цей файл dhcp.network. Таким чином шляхом цього файлу є /etc/systemd/network/dhcp.network. Вміст файлу наведено нижче

```
[Match]
Name=en*

[Network]
Description=Enable DHCP client for each ethernet interfaces
DHCP=yes
```

Параметр Name має значення en*. Завдяки цьому ця конфігурація застосовується для всіх мережевих інтерфейсів, назви яких починаються з en. Це дозволяє застосувати її до всіх інтерфейсів Ethernet, відповідно до схеми іменування мережевих пристроїв [39], при цьому ігноруючи інтерфейс lo (loopback інтерфейс). Активація служби DHCP клієнта відбувається за допомогою параметра DHCP зі значенням yes. Це значення дозволяє активувати DHCP клієнт для IPv4 та IPv6.

Активуємо автозапуск служби systemd-networkd [22].

```
[root@hostname /]# systemctl enable systemd-networkd.service
```

Таким чином система, під час запуску, автоматично запитуватиме адресу та інші мережеві параметри у DHCP сервера.

Також маємо активувати службу розв'язання доменних імен.

```
[root@hostname /]# systemctl enable systemd-resolved.service
```

В.5.5. Додаткові налаштування

Налаштовуємо ряд додаткових служб і параметрів:

— ім'я хосту "admin-usb відповідно до додатка Л;
— локаль, відповідно до додатка М;
— віддалений доступ, відповідно до додатка Н;
— брандмауер, відповідно до розділу 2.9;
— пароль адміністратора, відповідно до розділу 2.10.

Встановлюємо ряд допоміжних утиліт:

— manjaro-tools. Набір утиліт від розробників Manjaro. Завдяки яким значно спрощується взаємодія з системою я якості адміністратора.

— rsync [40]. Система дистанційної синхронізації файлів. Вона є дуже гнучкою та підтримує дуже багато режимів та параметрів;
— dosfstools. Набір утиліт для створення файлових систем MS-DOS.

```
[root@hostname /]# pacman -Syyu rsync  manjaro-tools-{base,pkg}-git dosfstools
```

На цьому налаштування образу системи можна завершити. Виходимо з віртуального середовища за допомогою команди exit.

```
[root@hostname /]# exit
```

В.6. Копіювання образу

Під'єднуємо носій до файлової системи комп'ютера. Монтування виконуватимемо відповідно до налаштувань у таблиці В.3. Для цього використаємо утиліту mount [41] з ключем mkdir і вказуємо їх шлях до файлу пристрою і точку монтування. Для під'єднання виконуємо наступні команди.

```
[root@hostname ~]# mount --mkdir /dev/sdb3 /mnt/usb
[root@hostname ~]# mount --mkdir /dev/sdb2 /mnt/usb/efi
```

Для перевірки монтування знову скористаємось програмою mount, але на цей раз без параметрів і відфільтруємо вивід по назві файлу пристрою.

```
[root@hostname ~]# mount | grep sdb
/dev/sdb3 on /mnt/usb type ext4 (rw,relatime)
/dev/sdb2 on /mnt/usb/efi type vfat (rw,relatime,fmask=0022,dmask=0022,
↪  codepage=437,iocharset=ascii,shortname=mixed,utf8,errors=remount-ro)
```

Як бачимо, пристрій коректно примонтований. Тепер скопіюємо сконфігурований образ системи на пристрій. Для копіювання скористаємось програмою cp [42] з ключем archive, вказавши шлях до образу та шлях до точки монтування пристрою.

```
[root@hostname ~]# cp --archive /home/bogdan/USB_OS/* /mnt/usb/
```

Використання зірочки є обов'язковим, інакше скопіюється сама директорія, а не її вміст.

Після завершення процесу копіювання, перевіряємо скопійовані файли за допомогою команди ls [43] з ключами l та a, передавши шлях до точки монтування.

```
[root@hostname ~]# ls -al /mnt/usb/
загалом 64
drwxr-xr-x 16 root root 4096 кві 28 15:55 .
drwxr-xr-x  3 root root 4096 кві 28 15:14 ..
lrwxrwxrwx  1 root root    7 бер 21 16:24 bin -> usr/bin
drwxr-xr-x  3 root root 4096 січ  1  1970 boot
drwxr-xr-x  2 root root 4096 кві 22 20:40 dev
drwxr-xr-x 37 root root 4096 кві 23 11:42 etc
drwxr-xr-x  2 root root 4096 бер 21 16:24 home
lrwxrwxrwx  1 root root    7 бер 21 16:24 lib -> usr/lib
lrwxrwxrwx  1 root root    7 бер 21 16:24 lib64 -> usr/lib
drwxr-xr-x  2 root root 4096 бер 21 16:24 mnt
drwxr-xr-x  2 root root 4096 бер 21 16:24 opt
dr-xr-xr-x  2 root root 4096 кві 22 20:40 proc
drwxr-x---  4 root root 4096 кві 23 10:12 root
drwxr-xr-x  2 root root 4096 кві 22 20:40 run
lrwxrwxrwx  1 root root    7 бер 21 16:24 sbin -> usr/bin
drwxr-xr-x  4 root root 4096 кві 22 20:41 srv
dr-xr-xr-x  2 root root 4096 кві 22 20:40 sys
drwxrwxrwt  2 root root 4096 кві 22 20:40 tmp
drwxr-xr-x  8 root root 4096 кві 23 11:42 usr
drwxr-xr-x 12 root root 4096 кві 22 20:41 var
```

В.7. Встановлення завантажувача

Для встановлення завантажувача треба знову створити й увійти у віртуальне середовище з портативною системою. Але на цього разу зі встановленою. Для цього використовуємо утиліту manjaro-chroot передавши їх точку монтування пристрою.

```
[root@hostname ~]# manjaro-chroot /mnt/usb/ /bin/bash
```

Тепер встановімо завантажувач GRUB. Скориставшись програмою grub-install [20]. Встановлюємо завантажувач для систем BIOS за допомогою команди.

```
[root@hostname /]# grub-install --target=i386-pc /dev/sdb
```

І встановлюємо завантажувач для систем UEFI за допомогою команди.

```
[root@hostname /]# grub-install --target=x86_64-efi --removable
↪  --efi-directory=/efi /dev/sdb
```

Тепер будуємо завантажувальний образ системи та острів ядра системи. Також оновлюємо конфігурації завантажувача. Для генерації образу скористаємося системою збірки mkinitcpio [16], котру вже налаштовано у розділі В.5.3.1. Для генерації образу треба викликати програму mkinitcpio з ключем allpresets.

```
[root@hostname /]# mkinitcpio --allpresets
```

Оновлюємо конфігурації завантажувача за допомогою утиліти update-grub.

```
[root@hostname /]# update-grub
```

Виходимо із віртуального середовища.

```
[root@hostname /]# exit
```

Від’єднуємо пристрій за допомогою утиліти umount [44] і видаляємо точку монтування за допомогою rmdir [45].

```
[root@hostname ~]# umount -R /mnt/usb
[root@hostname ~]# rm -r /mnt/usb
```

ДОДАТОК Г
ІНСТРУКЦІЯ З ВИКОРИСТАННЯ ПРОГРАМИ FDISK

Г.1. Загальна інформація

fdisk — це діалогова утиліта командного рядка, яка створює таблиці розділів і розділи на жорсткому диску та керує ними.

Програма має 2 режими роботи:

— автоматичний;

— діалоговий.

Автоматичний режим використовується для резервування, відновлення і копіювання налаштувань диску.

Діалоговий режим використовується для конфігурування дискового простору.

Для запуску програми в діалоговому режимі треба запустити fdisk вказавши шлях до файлу пристрою.

```
fdisk /dev/sda
```

Програма зустрічає користувача привітанням та запитом команди.

```
Вітаємо у fdisk (util-linux 2.38.1).
Зміни буде збережено у пам'яті, доки ви не вирішите записати їх.
Будьте обережні з використанням команди запису.

Цей диск наразі використовується - зміна розділів, можливо, погана ідея.
Рекомендуємо демонтувати всі файлові системи та вимкнути всі розділи свопінгу
на цьому диску.

Команда (m - довідка):
```

Повний перелік команд з описами доступний у довідці. Для показу довідки необхідно ввести команду m.

```
Команда (m - довідка): m

Довідка:

  DOS (MBR)
   a   перемкнути прапорець завантажуваності
```

```
 b   змінити вкладену мітку диска BSD
 c   перемкнути прапорець сумісності з dos

Загальне
 d   вилучити розділ
 F   показати список нерозподіленого місця
 l   показати список відомих типів розділів
 n   додати новий розділ
 p   вивести таблицю розділів
 t   змінити тип розділу
 v   перевірити таблицю розділів
 i   вивести дані щодо розділу

Інше
 m   вивести це меню
 u   змінити одиниці показу/введення
 x   додаткові функціональні можливості (лише для фахівців)

Скрипт
 I   завантажити компонування диска із файла скрипту sfdisk
 O   створити дамп компонування диска до файла скрипту sfdisk

Зберегти і вийти
 w   записати таблицю на диск і вийти
 q   вийти без збереження змін

Створити нову мітку
 g   створити нову порожню таблицю розділів GPT
 G   створити нову порожню таблицю розділів SGI (IRIX)
 o   створити нову порожню таблицю розділів DOS
 s   створити нову порожню таблицю розділів Sun
```

Г.2. Створення нової таблиці розділів

fdisk підтримує роботу з 4-ма таблицями розділів:

— GPT;
— SGI (IRIX);
— DOS (Master Boot Record (MBR));
— Sun.

Для створення нової таблиці розділів GPT необхідно виконати команду g.

Г.3. Створення нового розділу

Для створення використовується команда n. Вона запитає номер розділу, перший сектор і останній сектор розділу. Ниже наведено приклад створення розділу об'ємом 8 MiB на початку дискового простору.

```
Команда (m - довідка): n
Номер розділу (1-128, типово 1):
Перший сектор (2048-60628958, типово 2048):
Останній сектор, +/-sectors або +/-size{K,M,G,T,P} (2048-60628958, типово
↪  60626943): +8M

Створено новий розділ 1 типу «Linux filesystem», розмір - 8 MiB.
```

Г.4. Зміна типу розділу

Тип розділу — це набір специфічних для розділу параметрів. Тим може декларувати призначення, формат, структуру розділу та багато іншого. За замовчуванням fdisk створює розділи з типом Linux filesystem.

Для виводу повного переліку типів розділу необхідно виконати команду l. Його наведено нижче.

```
 1 EFI System                     C12A7328-F81F-11D2-BA4B-00A0C93EC93B
 2 MBR partition scheme           024DEE41-33E7-11D3-9D69-0008C781F39F
 3 Intel Fast Flash               D3BFE2DE-3DAF-11DF-BA40-E3A556D89593
 4 BIOS boot                      21686148-6449-6E6F-744E-656564454649
 5 Sony boot partition            F4019732-066E-4E12-8273-346C5641494F
 6 Lenovo boot partition          BFBFAFE7-A34F-448A-9A5B-6213EB736C22
 7 PowerPC PReP boot              9E1A2D38-C612-4316-AA26-8B49521E5A8B
 8 ONIE boot                      7412F7D5-A156-4B13-81DC-867174929325
 9 ONIE config                    D4E6E2CD-4469-46F3-B5CB-1BFF57AFC149
10 Microsoft reserved             E3C9E316-0B5C-4DB8-817D-F92DF00215AE
11 Microsoft basic data           EBD0A0A2-B9E5-4433-87C0-68B6B72699C7
12 Microsoft LDM metadata         5808C8AA-7E8F-42E0-85D2-E1E90434CFB3
13 Microsoft LDM data             AF9B60A0-1431-4F62-BC68-3311714A69AD
14 Windows recovery environment   DE94BBA4-06D1-4D40-A16A-BFD50179D6AC
15 IBM General Parallel Fs        37AFFC90-EF7D-4E96-91C3-2D7AE055B174
16 Microsoft Storage Spaces       E75CAF8F-F680-4CEE-AFA3-B001E56EFC2D
17 HP-UX data                     75894C1E-3AEB-11D3-B7C1-7B03A0000000
18 HP-UX service                  E2A1E728-32E3-11D6-A682-7B03A0000000
19 Linux swap                     0657FD6D-A4AB-43C4-84E5-0933C84B4F4F
20 Linux filesystem               0FC63DAF-8483-4772-8E79-3D69D8477DE4
21 Linux server data              3B8F8425-20E0-4F3B-907F-1A25A76F98E8
```

```
22 Linux root (x86)               44479540-F297-41B2-9AF7-D131D5F0458A
23 Linux root (x86-64)            4F68BCE3-E8CD-4DB1-96E7-FBCAF984B709
24 Linux root (Alpha)             6523F8AE-3EB1-4E2A-A05A-18B695AE656F
25 Linux root (ARC)               D27F46ED-2919-4CB8-BD25-9531F3C16534
26 Linux root (ARM)               69DAD710-2CE4-4E3C-B16C-21A1D49ABED3
27 Linux root (ARM-64)            B921B045-1DF0-41C3-AF44-4C6F280D3FAE
28 Linux root (IA-64)             993D8D3D-F80E-4225-855A-9DAF8ED7EA97
29 Linux root (LoongArch-64)      77055800-792C-4F94-B39A-98C91B762BB6
30 Linux root (MIPS-32 LE)        37C58C8A-D913-4156-A25F-48B1B64E07F0
31 Linux root (MIPS-64 LE)        700BDA43-7A34-4507-B179-EEB93D7A7CA3
32 Linux root (PPC)               1DE3F1EF-FA98-47B5-8DCD-4A860A654D78
33 Linux root (PPC64)             912ADE1D-A839-4913-8964-A10EEE08FBD2
34 Linux root (PPC64LE)           C31C45E6-3F39-412E-80FB-4809C4980599
35 Linux root (RISC-V-32)         60D5A7FE-8E7D-435C-B714-3DD8162144E1
36 Linux root (RISC-V-64)         72EC70A6-CF74-40E6-BD49-4BDA08E8F224
37 Linux root (S390)              08A7ACEA-624C-4A20-91E8-6E0FA67D23F9
38 Linux root (S390X)             5EEAD9A9-FE09-4A1E-A1D7-520D00531306
39 Linux root (TILE-Gx)           C50CDD70-3862-4CC3-90E1-809A8C93EE2C
40 Linux reserved                 8DA63339-0007-60C0-C436-083AC8230908
41 Linux home                     933AC7E1-2EB4-4F13-B844-0E14E2AEF915
42 Linux RAID                     A19D880F-05FC-4D3B-A006-743F0F84911E
43 Linux LVM                      E6D6D379-F507-44C2-A23C-238F2A3DF928
44 Linux variable data            4D21B016-B534-45C2-A9FB-5C16E091FD2D
45 Linux temporary data           7EC6F557-3BC5-4ACA-B293-16EF5DF639D1
46 Linux /usr (x86)               75250D76-8CC6-458E-BD66-BD47CC81A812
47 Linux /usr (x86-64)            8484680C-9521-48C6-9C11-B0720656F69E
48 Linux /usr (Alpha)             E18CF08C-33EC-4C0D-8246-C6C6FB3DA024
49 Linux /usr (ARC)               7978A683-6316-4922-BBEE-38BFF5A2FECC
50 Linux /usr (ARM)               7D0359A3-02B3-4F0A-865C-654403E70625
51 Linux /usr (ARM-64)            B0E01050-EE5F-4390-949A-9101B17104E9
52 Linux /usr (IA-64)             4301D2A6-4E3B-4B2A-BB94-9E0B2C4225EA
53 Linux /usr (LoongArch-64)      E611C702-575C-4CBE-9A46-434FA0BF7E3F
54 Linux /usr (MIPS-32 LE)        0F4868E9-9952-4706-979F-3ED3A473E947
55 Linux /usr (MIPS-64 LE)        C97C1F32-BA06-40B4-9F22-236061B08AA8
56 Linux /usr (PPC)               7D14FEC5-CC71-415D-9D6C-06BF0B3C3EAF
57 Linux /usr (PPC64)             2C9739E2-F068-46B3-9FD0-01C5A9AFBCCA
58 Linux /usr (PPC64LE)           15BB03AF-77E7-4D4A-B12B-C0D084F7491C
59 Linux /usr (RISC-V-32)         B933FB22-5C3F-4F91-AF90-E2BB0FA50702
60 Linux /usr (RISC-V-64)         BEAEC34B-8442-439B-A40B-984381ED097D
61 Linux /usr (S390)              CD0F869B-D0FB-4CA0-B141-9EA87CC78D66
62 Linux /usr (S390X)             8A4F5770-50AA-4ED3-874A-99B710DB6FEA
63 Linux /usr (TILE-Gx)           55497029-C7C1-44CC-AA39-815ED1558630
64 Linux root verity (x86)        D13C5D3B-B5D1-422A-B29F-9454FDC89D76
65 Linux root verity (x86-64)     2C7357ED-EBD2-46D9-AEC1-23D437EC2BF5
66 Linux root verity (Alpha)      FC56D9E9-E6E5-4C06-BE32-E74407CE09A5
66 Linux root verity (Alpha)      FC56D9E9-E6E5-4C06-BE32-E74407CE09A5
67 Linux root verity (ARC)        24B2D975-0F97-4521-AFA1-CD531E421B8D
68 Linux root verity (ARM)        7386CDF2-203C-47A9-A498-F2ECCE45A2D6
```

```
 69 Linux root verity (ARM-64)     DF3300CE-D69F-4C92-978C-9BFB0F38D820
 70 Linux root verity (IA-64)      86ED10D5-B607-45BB-8957-D350F23D0571
 71 Linux root verity (LoongArch-64) F3393B22-E9AF-4613-A948-9D3BFBD0C535
 72 Linux root verity (MIPS-32 LE) D7D150D2-2A04-4A33-8F12-16651205FF7B
 73 Linux root verity (MIPS-64 LE) 16B417F8-3E06-4F57-8DD2-9B5232F41AA6
 74 Linux root verity (PPC)        98CFE649-1588-46DC-B2F0-ADD147424925
 75 Linux root verity (PPC64)      9225A9A3-3C19-4D89-B4F6-EEFF88F17631
 76 Linux root verity (PPC64LE)    906BD944-4589-4AAE-A4E4-DD983917446A
 77 Linux root verity (RISC-V-32)  AE0253BE-1167-4007-AC68-43926C14C5DE
 78 Linux root verity (RISC-V-64)  B6ED5582-440B-4209-B8DA-5FF7C419EA3D
 79 Linux root verity (S390)       7AC63B47-B25C-463B-8DF8-B4A94E6C90E1
 80 Linux root verity (S390X)      B325BFBE-C7BE-4AB8-8357-139E652D2F6B
 81 Linux root verity (TILE-Gx)    966061EC-28E4-4B2E-B4A5-1F0A825A1D84
 82 Linux /usr verity (x86)        8F461B0D-14EE-4E81-9AA9-049B6FB97ABD
 83 Linux /usr verity (x86-64)     77FF5F63-E7B6-4633-ACF4-1565B864C0E6
 84 Linux /usr verity (Alpha)      8CCE0D25-C0D0-4A44-BD87-46331BF1DF67
 85 Linux /usr verity (ARC)        FCA0598C-D880-4591-8C16-4EDA05C7347C
 86 Linux /usr verity (ARM)        C215D751-7BCD-4649-BE90-6627490A4C05
 87 Linux /usr verity (ARM-64)     6E11A4E7-FBCA-4DED-B9E9-E1A512BB664E
 88 Linux /usr verity (IA-64)      6A491E03-3BE7-4545-8E38-83320E0EA880
 89 Linux /usr verity (LoongArch-64) F46B2C26-59AE-48F0-9106-C50ED47F673D
 90 Linux /usr verity (MIPS-32 LE) 46B98D8D-B55C-4E8F-AAB3-37FCA7F80752
 91 Linux /usr verity (MIPS-64 LE) 3C3D61FE-B5F3-414D-BB71-8739A694A4EF
 92 Linux /usr verity (PPC)        DF765D00-270E-49E5-BC75-F47BB2118B09
 93 Linux /usr verity (PPC64)      BDB528A5-A259-475F-A87D-DA53FA736A07
 94 Linux /usr verity (PPC64LE)    EE2B9983-21E8-4153-86D9-B6901A54D1CE
 95 Linux /usr verity (RISC-V-32)  CB1EE4E3-8CD0-4136-A0A4-AA61A32E8730
 96 Linux /usr verity (RISC-V-64)  8F1056BE-9B05-47C4-81D6-BE53128E5B54
 97 Linux /usr verity (S390)       B663C618-E7BC-4D6D-90AA-11B756BB1797
 98 Linux /usr verity (S390X)      31741CC4-1A2A-4111-A581-E00B447D2D06
 99 Linux /usr verity (TILE-Gx)    2FB4BF56-07FA-42DA-8132-6B139F2026AE
100 Linux root verity sign. (x86)  5996FC05-109C-48DE-808B-23FA0830B676
101 Linux root verity sign. (x86-64) 41092B05-9FC8-4523-994F-2DEF0408B176
102 Linux root verity sign. (Alpha) D46495B7-A053-414F-80F7-700C99921EF8
103 Linux root verity sign. (ARC)  143A70BA-CBD3-4F06-919F-6C05683A78BC
104 Linux root verity sign. (ARM)  42B0455F-EB11-491D-98D3-56145BA9D037
105 Linux root verity sign. (ARM-64) 6DB69DE6-29F4-4758-A7A5-962190F00CE3
106 Linux root verity sign. (IA-64) E98B36EE-32BA-4882-9B12-0CE14655F46A
107 Linux root verity sign. (LoongArch-64) 5AFB67EB-ECC8-4F85-AE8E-AC1E7C50E7D0
108 Linux root verity sign. (MIPS-32 LE) C919CC1F-4456-4EFF-918C-F75E94525CA5
109 Linux root verity sign. (MIPS-64 LE) 904E58EF-5C65-4A31-9C57-6AF5FC7C5DE7
110 Linux root verity sign. (PPC)  1B31B5AA-ADD9-463A-B2ED-BD467FC857E7
111 Linux root verity sign. (PPC64) F5E2C20C-45B2-4FFA-BCE9-2A60737E1AAF
112 Linux root verity sign. (PPC64LE) D4A236E7-E873-4C07-BF1D-BF6CF7F1C3C6
113 Linux root verity sign. (RISC-V-32) 3A112A75-8729-4380-B4CF-764D79934448
114 Linux root verity sign. (RISC-V-64) EFE0F087-EA8D-4469-821A-4C2A96A8386A
115 Linux root verity sign. (S390) 3482388E-4254-435A-A241-766A065F9960
116 Linux root verity sign. (S390X) C80187A5-73A3-491A-901A-017C3FA953E9
```

```
117 Linux root verity sign. (TILE-Gx) B3671439-97B0-4A53-90F7-2D5A8F3AD47B
118 Linux /usr verity sign. (x86)  974A71C0-DE41-43C3-BE5D-5C5CCD1AD2C0
119 Linux /usr verity sign. (x86-64) E7BB33FB-06CF-4E81-8273-E543B413E2E2
120 Linux /usr verity sign. (Alpha) 5C6E1C76-076A-457A-A0FE-F3B4CD21CE6E
121 Linux /usr verity sign. (ARC)  94F9A9A1-9971-427A-A400-50CB297F0F35
122 Linux /usr verity sign. (ARM)  D7FF812F-37D1-4902-A810-D76BA57B975A
123 Linux /usr verity sign. (ARM-64) C23CE4FF-44BD-4B00-B2D4-B41B3419E02A
124 Linux /usr verity sign. (IA-64) 8DE58BC2-2A43-460D-B14E-A76E4A17B47F
125 Linux /usr verity sign. (LoongArch-64) B024F315-D330-444C-8461-44BBDE524E99
126 Linux /usr verity sign. (MIPS-32 LE) 3E23CA0B-A4BC-4B4E-8087-5AB6A26AA8A9
127 Linux /usr verity sign. (MIPS-64 LE) F2C2C7EE-ADCC-4351-B5C6-EE9816B66E16
128 Linux /usr verity sign. (PPC)  7007891D-D371-4A80-86A4-5CB875B9302E
129 Linux /usr verity sign. (PPC64) 0B888863-D7F8-4D9E-9766-239FCE4D58AF
130 Linux /usr verity sign. (PPC64LE) C8BFBD1E-268E-4521-8BBA-BF314C399557
131 Linux /usr verity sign. (RISC-V-32) C3836A13-3137-45BA-B583-B16C50FE5EB4
132 Linux /usr verity sign. (RISC-V-64) D2F9000A-7A18-453F-B5CD-4D32F77A7B32
133 Linux /usr verity sign. (S390) 17440E4F-A8D0-467F-A46E-3912AE6EF2C5
134 Linux /usr verity sign. (S390X) 3F324816-667B-46AE-86EE-9B0C0C6C11B4
135 Linux /usr verity sign. (TILE-Gx) 4EDE75E2-6CCC-4CC8-B9C7-70334B087510
136 Linux extended boot            BC13C2FF-59E6-4262-A352-B275FD6F7172
137 Linux user's home              773f91ef-66d4-49b5-bd83-d683bf40ad16
138 FreeBSD data                   516E7CB4-6ECF-11D6-8FF8-00022D09712B
139 FreeBSD boot                   83BD6B9D-7F41-11DC-BE0B-001560B84F0F
140 FreeBSD swap                   516E7CB5-6ECF-11D6-8FF8-00022D09712B
141 FreeBSD UFS                    516E7CB6-6ECF-11D6-8FF8-00022D09712B
142 FreeBSD ZFS                    516E7CBA-6ECF-11D6-8FF8-00022D09712B
143 FreeBSD Vinum                  516E7CB8-6ECF-11D6-8FF8-00022D09712B
144 Apple HFS/HFS+                 48465300-0000-11AA-AA11-00306543ECAC
145 Apple APFS                     7C3457EF-0000-11AA-AA11-00306543ECAC
146 Apple UFS                      55465300-0000-11AA-AA11-00306543ECAC
147 Apple RAID                     52414944-0000-11AA-AA11-00306543ECAC
148 Apple RAID offline             52414944-5F4F-11AA-AA11-00306543ECAC
149 Apple boot                     426F6F74-0000-11AA-AA11-00306543ECAC
150 Apple label                    4C616265-6C00-11AA-AA11-00306543ECAC
151 Apple TV recovery              5265636F-7665-11AA-AA11-00306543ECAC
152 Apple Core storage             53746F72-6167-11AA-AA11-00306543ECAC
153 Apple Silicon boot             69646961-6700-11AA-AA11-00306543ECAC
154 Apple Silicon recovery         52637672-7900-11AA-AA11-00306543ECAC
155 Solaris boot                   6A82CB45-1DD2-11B2-99A6-080020736631
156 Solaris root                   6A85CF4D-1DD2-11B2-99A6-080020736631
157 Solaris /usr & Apple ZFS       6A898CC3-1DD2-11B2-99A6-080020736631
158 Solaris swap                   6A87C46F-1DD2-11B2-99A6-080020736631
159 Solaris backup                 6A8B642B-1DD2-11B2-99A6-080020736631
160 Solaris /var                   6A8EF2E9-1DD2-11B2-99A6-080020736631
161 Solaris /home                  6A90BA39-1DD2-11B2-99A6-080020736631
162 Solaris alternate sector       6A9283A5-1DD2-11B2-99A6-080020736631
163 Solaris reserved 1             6A945A3B-1DD2-11B2-99A6-080020736631
```

```
164 Solaris reserved 2
↪  6A9630D1-1B626D89C-E1AE-C24E-9AE6-7CA5899088A2DD2-11B2-99A6-080020736631
165 Solaris reserved 3              6A980767-1DD2-11B2-99A6-080020736631
166 Solaris reserved 4              6A96237F-1DD2-11B2-99A6-080020736631
167 Solaris reserved 5              6A8D2AC7-1DD2-11B2-99A6-080020736631
168 NetBSD swap                     49F48D32-B10E-11DC-B99B-0019D1879648
169 NetBSD FFS                      49F48D5A-B10E-11DC-B99B-0019D1879648
170 NetBSD LFS                      49F48D82-B10E-11DC-B99B-0019D1879648
171 NetBSD concatenated             2DB519C4-B10F-11DC-B99B-0019D1879648
172 NetBSD encrypted                2DB519EC-B10F-11DC-B99B-0019D1879648
173 NetBSD RAID                     49F48DAA-B10E-11DC-B99B-0019D1879648
174 ChromeOS kernel                 FE3A2A5D-4F32-41A7-B725-ACCC3285A309
175 ChromeOS root fs                3CB8E202-3B7E-47DD-8A3C-7FF2A13CFCEC
176 ChromeOS reserved               2E0A753D-9E48-43B0-8337-B15192CB1B5E
177 MidnightBSD data                85D5E45A-237C-11E1-B4B3-E89A8F7FC3A7
178 MidnightBSD boot                85D5E45E-237C-11E1-B4B3-E89A8F7FC3A7
179 MidnightBSD swap                85D5E45B-237C-11E1-B4B3-E89A8F7FC3A7
180 MidnightBSD UFS                 0394EF8B-237E-11E1-B4B3-E89A8F7FC3A7
181 MidnightBSD ZFS                 85D5E45D-237C-11E1-B4B3-E89A8F7FC3A7
182 MidnightBSD Vinum               85D5E45C-237C-11E1-B4B3-E89A8F7FC3A7
183 Ceph Journal                    45B0969E-9B03-4F30-B4C6-B4B80CEFF106
184 Ceph Encrypted Journal          45B0969E-9B03-4F30-B4C6-5EC00CEFF106
185 Ceph OSD                        4FBD7E29-9D25-41B8-AFD0-062C0CEFF05D
186 Ceph crypt OSD                  4FBD7E29-9D25-41B8-AFD0-5EC00CEFF05D
187 Ceph disk in creation           89C57F98-2FE5-4DC0-89C1-F3AD0CEFF2BE
188 Ceph crypt disk in creation     89C57F98-2FE5-4DC0-89C1-5EC00CEFF2BE
189 VMware VMFS                     AA31E02A-400F-11DB-9590-000C2911D1B8
190 VMware Diagnostic               9D275380-40AD-11DB-BF97-000C2911D1B8
191 VMware Virtual SAN              381CFCCC-7288-11E0-92EE-000C2911D0B2
192 VMware Virsto                   77719A0C-A4A0-11E3-A47E-000C29745A24
193 VMware Reserved                 9198EFFC-31C0-11DB-8F78-000C2911D1B8
194 OpenBSD data                    824CC7A0-36A8-11E3-890A-952519AD3F61
195 QNX6 file system                CEF5A9AD-73BC-4601-89F3-CDEEEEE321A1
196 Plan 9 partition                C91818F9-8025-47AF-89D2-F030D7000C2C
197 HiFive FSBL                     5B193300-FC78-40CD-8002-E86C45580B47
198 HiFive BBL                      2E54B353-1271-4842-806F-E436D6AF6985
199 Haiku BFS                       42465331-3BA3-10F1-802A-4861696B7521
200 Marvell Armada 3700 Boot partition 6828311A-BA55-42A4-BCDE-A89BB5EDECAE

Варіанти:
   linux          - 0FC63DAF-8483-4772-8E79-3D69D8477DE4
   swap           - 0657FD6D-A4AB-43C4-84E5-0933C84B4F4F
   home           - 933AC7E1-2EB4-4F13-B844-0E14E2AEF915
   uefi           - C12A7328-F81F-11D2-BA4B-00A0C93EC93B
   raid           - A19D880F-05FC-4D3B-A006-743F0F84911E
   lvm            - E6D6D379-F507-44C2-A23C-238F2A3DF928
```

Для зміни типу розділу викликаємо команду t. Надаючи їй номер розділу для

редагування і новий тип розділу. Приклад нижче демонструє зміну розділу 1 на BIOS boot.

```
Команда (m - довідка): t
Вибраний розділ 1
Тип розділу або варіант (натисніть L, щоб переглянути список): 4
Тип розділу «Linux filesystem» змінено на «BIOS boot».
```

Г.5. Вивід таблиці розділів

Для показу поточної конфігурації дискового простору використовується команда p.

```
Команда (m - довідка): p

Диск /dev/sdb: 28,91 GiB, 31042043904 байтів, 60628992 секторів
Модель диска: USB FLASH DRIVE
Одиниці: секторів з 1 * 512 = 512 байтів
Розмір сектора (логічного/фізичного): 512 байтів / 512 байтів
Розмір введення-виведення (мінімальний/оптимальний): 512 байтів / 512 байтів
Тип мітки диска: gpt
Ідентифікатор диска: B626D89C-E1AE-C24E-9AE6-7CA5899088A2

Пристрій   Початок   Кінець  Сектори Розмір Тип
/dev/sdb1     2048    18431    16384     8M BIOS, завантажувальний
/dev/sdb2    18432  1067007  1048576   512M Система EFI
/dev/sdb3  1067008 60626943 59559936  28,4G Файлова система Linux
```

Г.6. Перевірка помилок

Для перевірки налаштувань використовується команда v. Ця команда перевірить наявну конфігурацію на наявність різноманітних помилок.

```
Команда (m - довідка): v

Помилок не виявлено.
Версія заголовка: 1.0
Використовуємо 3 з 128 розділів.
Загалом доступні 2015 вільних секторів у 1 фрагменті.
```

Г.7. Внесення змін

Під час роботи програма зберігає усі зміни в пам'яті. Завдяки цьому можна легко редагувати зміни не боячись про стан диску. Для внесення змін на диск використовується команда w.

```
Команда (m - довідка): w
Таблицю розділів було змінено.
Викликаємо ioctl(), щоб перечитати таблицю розділів.
Синхронізація дисків
```

Після успішного запису змін на носій, fdisk автоматично припиняє виконання.

ДОДАТОК Д
ІНСТРУКЦІЯ З ІНІЦІАЛІЗАЦІЇ ФАЙЛОВИХ СИСТЕМ

Д.1. Ext4

Для ініціалізації файлової системи Ext4 маємо викликати утиліту mkfs [35]. Надавши їй тип файлової системи ext4 і шлях до файлу розділу. Приклад відповідної команди наведено нижче.

```
[root@hostname ~]# mkfs -t ext4 /dev/sda1
```

Д.2. FAT

Для ініціалізації файлової системи FAT32 маємо викликати утиліту mkfs [35]. Надавши їй тип файлової системи fat і шлях до файлу розділу. Також бажано вказати параметр F з відповідною розрядністю: 12, 16, 32. Приклади відповідних команд наведено нижче.

FAT12:

```
[root@hostname ~]# mkfs -t fat -F 12 /dev/sda1
```

FAT16:

```
[root@hostname ~]# mkfs -t fat -F 16 /dev/sda1
```

FAT32:

```
[root@hostname ~]# mkfs -t fat -F 32 /dev/sda1
```

Д.3. SWAP

swap — це спеціальна файлова система призначена для розміщення сторінок пам'яті. Вона використовується для реалізації віртуальної пам'яті. Тобто в цих розділах зберігаються фрагменти пам'яті, що наразі не використовуються. Тим самим розвантажуючи оперативну пам'ять комп'ютера.

Також тут зберігається повний вміст оперативної пам'яті під час гібернації системи. Завдяки чому система може дуже швидко відновити свій минулий стан з того ж місця.

Для створення розділу swap використовуємо програму mkswap. Надавши їй шлях до файлу пристрою. Приклад відповідної команди наведено нижче.

```
[root@hostname ~]# mkswap /dev/sda1
```

ДОДАТОК Е
ВМІСТ ФАЙЛУ MKINITCPIO.CONF

```
# vim:set ft=sh
# MODULES
# The following modules are loaded before any boot hooks are
# run.  Advanced users may wish to specify all system modules
# in this array.  For instance:
#     MODULES=(usbhid xhci_hcd)
MODULES=()

# BINARIES
# This setting includes any additional binaries a given user may
# wish into the CPIO image.  This is run last, so it may be used to
# override the actual binaries included by a given hook
# BINARIES are dependency parsed, so you may safely ignore libraries
BINARIES=()

# FILES
# This setting is similar to BINARIES above, however, files are added
# as-is and are not parsed in any way.  This is useful for config files.
FILES=()

# HOOKS
# This is the most important setting in this file.  The HOOKS control the
# modules and scripts added to the image, and what happens at boot time.
# Order is important, and it is recommended that you do not change the
# order in which HOOKS are added.  Run 'mkinitcpio -H <hook name>' for
# help on a given hook.
# 'base' is _required_ unless you know precisely what you are doing.
# 'udev' is _required_ in order to automatically load modules
# 'filesystems' is _required_ unless you specify your fs modules in MODULES
# Examples:
##   This setup specifies all modules in the MODULES setting above.
##   No RAID, lvm2, or encrypted root is needed.
#    HOOKS=(base)
#
##   This setup will autodetect all modules for your system and should
##   work as a sane default
#    HOOKS=(base udev autodetect modconf block filesystems fsck)
#
##   This setup will generate a 'full' image which supports most systems.
##   No autodetection is done.
#    HOOKS=(base udev modconf block filesystems fsck)
#
##   This setup assembles a mdadm array with an encrypted root file system.
##   Note: See 'mkinitcpio -H mdadm_udev' for more information on RAID devices.
```

```
#    HOOKS=(base udev modconf keyboard keymap consolefont block mdadm_udev encrypt
↪  filesystems fsck)
#
##   This setup loads an lvm2 volume group.
#    HOOKS=(base udev modconf block lvm2 filesystems fsck)
#
##   NOTE: If you have /usr on a separate partition, you MUST include the
#    usr and fsck hooks.
HOOKS=(base udev autodetect modconf kms keyboard keymap consolefont block
↪  mdadm_udev resume filesystems fsck)

# COMPRESSION
# Use this to compress the initramfs image. By default, gzip compression
# is used. Use 'cat' to create an uncompressed image.
COMPRESSION="gzip"
#COMPRESSION="bzip2"
#COMPRESSION="lzma"
#COMPRESSION="xz"
#COMPRESSION="lzop"
#COMPRESSION="lz4"
#COMPRESSION="zstd"

# COMPRESSION_OPTIONS
# Additional options for the compressor
#COMPRESSION_OPTIONS=()

# MODULES_DECOMPRESS
# Decompress kernel modules during initramfs creation.
# Enable to speedup boot process, disable to save RAM
# during early userspace. Switch (yes/no).
MODULES_DECOMPRESS="no"
```

ДОДАТОК Ж
ВМІСТ ФАЙЛУ GRUB

```
# GRUB boot loader configuration

GRUB_DEFAULT=saved
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR="Manjaro"
GRUB_CMDLINE_LINUX_DEFAULT="resume=UUID=6fc4811b-7cb2-48ab-81c7-fe95284fb9e8
↪  quiet udev.log_priority=3"
GRUB_CMDLINE_LINUX=""

# Preload both GPT and MBR modules so that they are not missed
GRUB_PRELOAD_MODULES="part_gpt part_msdos"

# Uncomment to enable booting from LUKS encrypted devices
#GRUB_ENABLE_CRYPTODISK=y

# Set to 'countdown' or 'menu' to change timeout behavior,
# press ESC key to display menu.
GRUB_TIMEOUT_STYLE=menu

# Uncomment to use basic console
GRUB_TERMINAL_INPUT=console

# Uncomment to disable graphical terminal
#GRUB_TERMINAL_OUTPUT=console

# The resolution used on graphical terminal
# note that you can use only modes which your graphic card supports via VBE
# you can see them in real GRUB with the command 'videoinfo'
GRUB_GFXMODE=auto

# Uncomment to allow the kernel use the same resolution used by grub
GRUB_GFXPAYLOAD_LINUX=keep

# Uncomment if you want GRUB to pass to the Linux kernel the old parameter
# format "root=/dev/xxx" instead of "root=/dev/disk/by-uuid/xxx"
#GRUB_DISABLE_LINUX_UUID=true

# Uncomment to disable generation of recovery mode menu entries
GRUB_DISABLE_RECOVERY=false

# Uncomment and set to the desired menu colors.  Used by normal and wallpaper
# modes only.  Entries specified as foreground/background.
GRUB_COLOR_NORMAL="light-gray/black"
GRUB_COLOR_HIGHLIGHT="green/black"
```

```
# Uncomment one of them for the gfx desired, a image background or a gfxtheme
#GRUB_BACKGROUND="/usr/share/grub/background.png"
GRUB_THEME="/usr/share/grub/themes/manjaro/theme.txt"

# Uncomment to get a beep at GRUB start
#GRUB_INIT_TUNE="480 440 1"

# Uncomment to make GRUB remember the last selection. This requires
# setting 'GRUB_DEFAULT=saved' above.
GRUB_SAVEDEFAULT=true

# Uncomment to disable submenus in boot menu
#GRUB_DISABLE_SUBMENU=y

# Uncomment this option to enable os-prober execution in the grub-mkconfig command
GRUB_DISABLE_OS_PROBER=true

# Uncomment to ensure that the root filesystem is mounted read-only so that
# systemd-fsck can run the check automatically. We use 'fsck' by default, which
# needs 'rw' as boot parameter, to avoid delay in boot-time. 'fsck' needs to be
# removed from 'mkinitcpio.conf' to make 'systemd-fsck' work.
# See also Arch-Wiki: https://wiki.archlinux.org/index.php/Fsck#Boot_time_checking
#GRUB_ROOT_FS_RO=true
```

ДОДАТОК И
ВМІСТ ФАЙЛУ FSTAB

```
# Static information about the filesystems.
# See fstab(5) for details.

# <file system> <dir> <type> <options> <dump> <pass>
UUID=be48508f-45a1-462c-85d8-368c7086a657       /       ext4    defaults
↪  0       1
UUID=0879-0B5E                                  /efi    vfat
↪  defaults,fat=32,umask=0077      0       2
UUID=cd4b8675-362e-4874-8dec-ea7692946e27       /var    ext4    defaults
↪  0       2
UUID=6fc4811b-7cb2-48ab-81c7-fe95284fb9e8       none    swap    defaults
↪  0       0
tmpfs                                           /tmp    tmpfs
↪  defaults,size=8G,mode=1777      0       0
```

ДОДАТОК К
БАЗОВИЙ НАБІР ПРАВИЛ БРАНДМАУЕРА

```
#!/usr/bin/nft -f
# vim:set ts=4 sw=4 et:

# IPv4/IPv6 Simple & Safe firewall ruleset.
# More examples in /usr/share/nftables/ and /usr/share/doc/nftables/examples/.

# Видалення старих правил
flush ruleset;

# Таємна послідовність
define KNOCK_0 = {ВИПАДКОВИЙ_ПОРТ};
define KNOCK_1 = {ВИПАДКОВИЙ_ПОРТ};
define KNOCK_2 = {ВИПАДКОВИЙ_ПОРТ};
define KNOCK_3 = {ВИПАДКОВИЙ_ПОРТ};
define KNOCK_4 = {ВИПАДКОВИЙ_ПОРТ};

# Таємні порти
define GUARDED_PORTS = {
    tcp . 22,
}

table inet filter {

    # Перелік кандидатів
    set candidates_ip4 {
        type ipv4_addr . inet_service;
        flags timeout;
        timeout 1s;
    }
    set candidates_ip6 {
        type ipv6_addr . inet_service;
        flags timeout;
        timeout 1s;
    }

    # Перелік клієнтів
    set clients_ip4 {
        type ipv4_addr;
        flags timeout;
        timeout 10s;
    }
    set clients_ip6 {
        type ipv6_addr;
        flags timeout;
```

```
    timeout 10s;
}

chain knocking_secret {
    # Надати доступ до таємних портів
    tcp dport $KNOCK_4 ip saddr . tcp dport @candidates_ip4 add @clients_ip4 {
    ↪  ip saddr };
    tcp dport $KNOCK_4 ip6 saddr . tcp dport @candidates_ip6 add @clients_ip6
    ↪  { ip6 saddr };

    # Проміжні етапи послідовності
    tcp dport $KNOCK_3 ip saddr . tcp dport @candidates_ip4 add
    ↪  @candidates_ip4 { ip saddr . $KNOCK_4 } return;
    tcp dport $KNOCK_3 ip6 saddr . tcp dport @candidates_ip6 add
    ↪  @candidates_ip6 { ip6 saddr . $KNOCK_4 } return;
    tcp dport $KNOCK_2 ip saddr . tcp dport @candidates_ip4 add
    ↪  @candidates_ip4 { ip saddr . $KNOCK_3 } return;
    tcp dport $KNOCK_2 ip6 saddr . tcp dport @candidates_ip6 add
    ↪  @candidates_ip6 { ip6 saddr . $KNOCK_3 } return;
    tcp dport $KNOCK_1 ip saddr . tcp dport @candidates_ip4 add
    ↪  @candidates_ip4 { ip saddr . $KNOCK_2 } return;
    tcp dport $KNOCK_1 ip6 saddr . tcp dport @candidates_ip6 add
    ↪  @candidates_ip6 { ip6 saddr . $KNOCK_2 } return;

    # Почати таємну послідовність
    tcp dport $KNOCK_0 add @candidates_ip4 { ip saddr . $KNOCK_1 } return;
    tcp dport $KNOCK_0 add @candidates_ip6 { ip6 saddr . $KNOCK_1 } return;

    # Видалити всі етапи кандидата у випадку помилки або проходження
    ↪  послідовності
    delete @candidates_ip4 { ip saddr . $KNOCK_0 };
    delete @candidates_ip6 { ip6 saddr . $KNOCK_0 };
    delete @candidates_ip4 { ip saddr . $KNOCK_1 };
    delete @candidates_ip6 { ip6 saddr . $KNOCK_1 };
    delete @candidates_ip4 { ip saddr . $KNOCK_2 };
    delete @candidates_ip6 { ip6 saddr . $KNOCK_2 };
    delete @candidates_ip4 { ip saddr . $KNOCK_3 };
    delete @candidates_ip6 { ip6 saddr . $KNOCK_3 };
    delete @candidates_ip4 { ip saddr . $KNOCK_4 };
    delete @candidates_ip6 { ip6 saddr . $KNOCK_4 };
}

# Перевірка таємної послідовності
chain knocking {
    # Почати перевірку таємної послідовності
    ct state new jump knocking_secret;

    # Надати доступ клієнтам до захищених портів
```

```
        meta l4proto . th dport $GUARDED_PORTS ip saddr @clients_ip4 accept;
        meta l4proto . th dport $GUARDED_PORTS ip6 saddr @clients_ip6 accept;
    }

    # Перерка вхідного трафіку
    chain input {
        type filter hook input priority filter; policy drop;

        # Дозволити зареєстровані з'єднання
        ct state vmap {
            established: accept,
            related: accept,
            invalid: drop,
        };

        # Дозволити локальні з'єднання
        iif lo accept;

        # Перевірити таємну послідовність
        jump knocking;
    }

    # Перевірка транзитного трафіку
    chain forward {
        type filter hook forward priority filter; policy drop;
    }

    # Перевірка вихідного трафіку
    chain output {
        type filter hook output priority filter; policy accept;
    }
}
```

ДОДАТОК Л
НАЛАШТУВАННЯ ІМЕНІ ХОСТУ

Налаштування імені комп'ютера в дистрибутивах з systemd відбувається шляхом редагування вмісту файлу /etc/hostname [46]. Для зміни назви бажано використовувати команду наступного виду.

```
[root@hostname /]# echo "{ІМ'Я_ХОСТУ}" > /etc/hostname
```

Де ІМ'Я_ХОСТУ це символ підстанови, що замінює ім'я комп'ютера.

Далі бажано відредагувати файл /etc/hosts [47] для коректної обробки назви хосту під час маршрутизації. Вміст типового файлу /etc/hosts [47] наведено нижче.

```
# Static table lookup for hostnames.
# See hosts(5) for details.

127.0.0.1    localhost
::1          localhost
127.0.1.1    {ІМ'Я_ХОСТУ}.eom.ust.edu.ua {ІМ'Я_ХОСТУ}
```

ДОДАТОК М
НАЛАШТУВАННЯ ЛОКАЛІ

М.1. Налаштування мови та шрифтів

Для коректної роботи частини протоколів, показу тексту та деяких інших елементів системи треба правильно налаштувати системну локаль.До поняття локаль відноситься набір налаштувань специфічних для певного регіону та/або мови.

Для правильного показу тексту в терміналі необхідно обрати консольний шрифт з підтримкою української мови. При використанні графічного інтерфейсу за показу тексту відповідає графічна система, але у портативній системі вона не потрібна. Тому цей аспект треба налаштовувати власноруч. Дізнаємось перелік доступних шрифтів за допомогою команди ls [43] вказавши шлях до директорії консольних штифтів. Також за допомогою команди grep [48] відфільтруємо шрифти з підтримкою Unicode.

```
[root@hostname /]# ls /usr/share/kbd/consolefonts/ | grep -i ".psfu.gz"
Agafari-12.psfu.gz
Agafari-14.psfu.gz
Agafari-16.psfu.gz
Cyr_a8x14.psfu.gz
Cyr_a8x16.psfu.gz
Cyr_a8x8.psfu.gz
Goha-12.psfu.gz
Goha-14.psfu.gz
Goha-16.psfu.gz
GohaClassic-12.psfu.gz
GohaClassic-14.psfu.gz
GohaClassic-16.psfu.gz
Lat2-Terminus16.psfu.gz
LatArCyrHeb-08.psfu.gz
LatArCyrHeb-14.psfu.gz
LatArCyrHeb-16+.psfu.gz
LatArCyrHeb-16.psfu.gz
LatArCyrHeb-19.psfu.gz
LatGrkCyr-12x22.psfu.gz
LatGrkCyr-8x16.psfu.gz
LatKaCyrHeb-14.psfu.gz
cp1250.psfu.gz
cp850-8x14.psfu.gz
cp850-8x16.psfu.gz
cp850-8x8.psfu.gz
```

```
cp865-8x14.psfu.gz
cp865-8x16.psfu.gz
cp865-8x8.psfu.gz
cyr-sun16.psfu.gz
default8x16.psfu.gz
default8x9.psfu.gz
drdos8x14.psfu.gz
drdos8x16.psfu.gz
drdos8x6.psfu.gz
drdos8x8.psfu.gz
eurlatgr.psfu.gz
gr737a-8x8.psfu.gz
gr737a-9x14.psfu.gz
gr737a-9x16.psfu.gz
gr737b-8x11.psfu.gz
gr737b-9x16-medieval.psfu.gz
gr737c-8x14.psfu.gz
gr737c-8x16.psfu.gz
gr737c-8x6.psfu.gz
gr737c-8x7.psfu.gz
gr737c-8x8.psfu.gz
gr737d-8x16.psfu.gz
gr928-8x16-thin.psfu.gz
gr928-9x14.psfu.gz
gr928-9x16.psfu.gz
gr928a-8x14.psfu.gz
gr928a-8x16.psfu.gz
gr928b-8x14.psfu.gz
gr928b-8x16.psfu.gz
greek-polytonic.psfu.gz
iso01-12x22.psfu.gz
iso02-12x22.psfu.gz
iso07u-16.psfu.gz
koi8r.8x8.psfu.gz
koi8u_8x14.psfu.gz
koi8u_8x16.psfu.gz
koi8u_8x8.psfu.gz
lat0-08.psfu.gz
lat0-10.psfu.gz
lat0-12.psfu.gz
lat0-14.psfu.gz
lat0-16.psfu.gz
lat0-sun16.psfu.gz
lat1-08.psfu.gz
lat1-10.psfu.gz
lat1-12.psfu.gz
lat1-14.psfu.gz
lat1-16.psfu.gz
```

```
lat2-08.psfu.gz
lat2-10.psfu.gz
lat2-12.psfu.gz
lat2-14.psfu.gz
lat2-16.psfu.gz
lat2-sun16.psfu.gz
lat2a-16.psfu.gz
lat4-08.psfu.gz
lat4-10.psfu.gz
lat4-12.psfu.gz
lat4-14.psfu.gz
lat4-16+.psfu.gz
lat4-16.psfu.gz
lat4-19.psfu.gz
lat4a-08.psfu.gz
lat4a-10.psfu.gz
lat4a-12.psfu.gz
lat4a-14.psfu.gz
lat4a-16+.psfu.gz
lat4a-16.psfu.gz
lat4a-19.psfu.gz
lat5-12.psfu.gz
lat5-14.psfu.gz
lat5-16.psfu.gz
lat7-14.psfu.gz
lat7a-14.psfu.gz
lat9u-08.psfu.gz
lat9u-10.psfu.gz
lat9u-12.psfu.gz
lat9u-14.psfu.gz
lat9u-16.psfu.gz
lat9v-08.psfu.gz
lat9v-10.psfu.gz
lat9v-12.psfu.gz
lat9v-14.psfu.gz
lat9v-16.psfu.gz
lat9w-08.psfu.gz
lat9w-10.psfu.gz
lat9w-12.psfu.gz
lat9w-14.psfu.gz
lat9w-16.psfu.gz
latarcyrheb-sun16.psfu.gz
latarcyrheb-sun32.psfu.gz
pancyrillic.f16.psfu.gz
ruscii_8x16.psfu.gz
ruscii_8x8.psfu.gz
solar24x32.psfu.gz
sun12x22.psfu.gz
```

```
viscii10-8x16.psfu.gz
```

З цього переліку найкраще підійдуть шрифти koi8u. Це шрифт Unicode з підтримкою україномовних символів. Цей шрифт постачається у трьох варіантах:

— 8x8;

— 8x14;

— 8x16.

Оберемо варіант 8x14. Цей варіант надає шрифт шириною 8 і висотою 14 пікселів.

Розкладку клавіатури за замовчуванням залишимо us (американська англійська).

Для зміни налаштувань відредагуємо файл /etc/vconsole.conf [49]. Відредагований файл має наступний вигляд.

```
KEYMAP=us
FONT=koi8u_8x14
```

Далі згенеруємо необхідні системні локалі. Для цього відредагуємо файл /etc/locale.gen [50]. Розкоментувавши локалі uk_UA.UTF-8 і uk_UA.KOI8-U та локаль en_US.UTF-8, як резервну. Відредагований вміст файлу виглядає наступним чином.

```
# Configuration file for locale-gen
#
# lists of locales that are to be generated by the locale-gen command.
#
# Each line is of the form:
#
#     <locale> <charset>
#
#  where <locale> is one of the locales given in /usr/share/i18n/locales
#  and <charset> is one of the character sets listed in /usr/share/i18n/charmaps
#
#  The locale-gen command will generate all the locales,
#  placing them in /usr/lib/locale.
#
#  A list of supported locales is given in /usr/share/i18n/SUPPORTED
#  and is included in this file. Uncomment the needed locales below.
#
#aa_DJ.UTF-8 UTF-8
# ...
```

```
en_US.UTF-8 UTF-8
# ...
uk_UA.UTF-8 UTF-8
uk_UA KOI8-U
# ...
#zu_ZA ISO-8859-1
```

Для генерації локалей скористаємось програмою locale-gen [51]. Вона автоматично згенерує вказані у файлі конфігурацій локалі.

```
[root@hostname /]# locale-gen
Generating locales...
  en_US.UTF-8... done
  uk_UA.UTF-8... done
  uk_UA.KOI8-U... done
Generation complete.
```

Тепер вказуємо системі локаль, що буде використовуватися. Для цього відредагуємо параметр LANG у файлі /etc/locale.conf [52].

```
LANG=uk_UA.UTF-8
```

М.2. Налаштування часового поясу

Для правильної роботи системного годинника та можливості автоматичної синхронізації часу, необхідно вказати часовий пояс. Інформація про часові пояси знаходиться у директорії /usr/share/zoneinfo та має ієрархічну структуру. Для налаштування часового поясу необхідно створити символічне посилання /etc/localtime [53] на файл з відповідною інформацією.

Для налаштування часового поясу Europe/Kyiv. Створюємо відповідне символічне посилання за допомогою програми ln [54].

```
[root@hostname /]# ln -sf /usr/share/zoneinfo/Europe/Kyiv /etc/localtime
```

ДОДАТОК Н
НАЛАШТУВАННЯ ВІДДАЛЕНОГО ДОСТУПУ

Н.1. Налаштування сервера

Для реалізації віддаленого доступу використовується протокол SSH. Для реалізації цієї служби обрано програмний комплекс openssh. Встановлюємо її за допомогою відповідного пакету.

```
[root@hostname /]# pacman -Syyu openssh
```

Спочатку бажано згенерувати нові ключові пари для сервера. Зазвичай це робиться автоматично, але для впевненості рекомендується. Генерація ключів відбувається аналогічно методу описаному в розділі Н.2, але без парольної фрази.

```
[root@hostname /]# ssh-keygen -t rsa -b 4096 -a 100 -f /etc/ssh/ssh_host_rsa_key
↪  -C "hostname"
[root@hostname /]# ssh-keygen -t ed25519 -a 100 -f /etc/ssh/ssh_host_ed25519_key
↪  -C "hostname"
```

Далі відредагуємо файл конфігурації сервера SSH. Він розміщений за шляхом /etc/ssh/sshd_config [55]. Для кожного сервера має бути власна конфігурація відповідно до його потреб та можливостей. Тому наведена нижче конфігурація є загальною.

```
# $OpenBSD: sshd_config,v 1.104 2021/07/02 05:11:21 dtucker Exp $

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/sbin:/usr/local/bin:/usr/bin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

Port 22
AddressFamily any
ListenAddress 0.0.0.0
ListenAddress ::
```

```
HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
RekeyLimit default none

# Logging
SyslogFacility AUTH
LogLevel INFO

# Authentication:

LoginGraceTime 30s
PermitRootLogin prohibit-password
StrictModes yes
MaxAuthTries 6
MaxSessions 10

PubkeyAuthentication yes

# The default is to check both .ssh/authorized_keys and .ssh/authorized_keys2
# but this is overridden so installations will only check .ssh/authorized_keys
AuthorizedKeysFile^^I.ssh/authorized_keys

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
IgnoreUserKnownHosts yes
# Don't read the user's ~/.rhosts and ~/.shosts files
IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication no
PermitEmptyPasswords no

# Change to no to disable s/key passwords
KbdInteractiveAuthentication no

# Kerberos options
KerberosAuthentication no
#KerberosOrLocalPasswd yes
```

```
#KerberosTicketCleanup yes
#KerberosGetAFSToken no

# GSSAPI options
GSSAPIAuthentication no
#GSSAPICleanupCredentials yes

# Set this to 'yes' to enable PAM authentication, account processing,
# and session processing. If this is enabled, PAM authentication will
# be allowed through the KbdInteractiveAuthentication and
# PasswordAuthentication.  Depending on your PAM configuration,
# PAM authentication via KbdInteractiveAuthentication may bypass
# the setting of "PermitRootLogin prohibit-password".
# If you just want the PAM account and session checks to run without
# PAM authentication, then enable this but set PasswordAuthentication
# and KbdInteractiveAuthentication to 'no'.
UsePAM yes

AllowAgentForwarding no
AllowTcpForwarding no
GatewayPorts no
X11Forwarding no
#X11DisplayOffset 10
#X11UseLocalhost yes
PermitTTY yes
PrintMotd no
PrintLastLog yes
TCPKeepAlive yes
PermitUserEnvironment no
Compression delayed
ClientAliveInterval 0
ClientAliveCountMax 3
UseDNS yes
PidFile /run/sshd.pid
MaxStartups 10:30:100
PermitTunnel no
ChrootDirectory none
VersionAddendum none

# no default banner path
#Banner none

# override default of no subsystems
Subsystem^^Isftp^^I/usr/lib/ssh/sftp-server

# Example of overriding settings on a per-user basis
Match User root
    AllowTcpForwarding yes
```

```
    AllowAgentForwarding no
    PermitTunnel yes
```

Після виконання всіх описаних вище дій, активуємо службу sshd [27].

```
[root@hostname /]# systemctl enable sshd.service
```

Н.2. Генерація ключів користувачів

Використання імені користувача та пароля для ідентифікації та автентифікації є поганим рішенням. Цей метод дуже вразливий до зламу. Тому в цілях забезпечення безпеки та зручності використовуватиметься, ідентифікація та автентифікація виконуватиметься за іменем користувача та ключем шифрування.

Для налаштування цього методу автентифікації необхідно згенерувати ключові пари. При генерації ключових пар враховані рекомендації з форумів [56—58]. Для облікових записів користувачів генеруємо дві ключові пари:

— ED25519. Цей тип ключа є найнадійнішим з представлених алгоритмів, але не всі реалізації SSH його підтримують. Це основна ключова;
— RSA. Цей тип ключа також вважається надійним. Для нього існують методи його зламу, але вони поки що не реалізовані та, можливо, не будуть реалізовані найближчим часом. Це резервна ключова пара на випадок якщо сервер не підтримує алгоритм ED25519.

Для більшої надійності вкажемо кожній парі унікальний пароль. Пароль забезпечить неможливість несанкціонованого використання ключів. Цей пароль є дуже важливим. Він має бути надійним та зберігатися в секреті.

Ключові пари генеруємо за допомогою програми ssh-keygen [59]. Для генерації перелічених ключових пар (адміністратора) виконуємо наступні команди.

```
[user@hostname ~]# ssh-keygen -t rsa -b 4096 -a 100 -C "user@eom.ust.edu.ua" -f
↪  ~/user_rsa
[user@hostname ~]# ssh-keygen -t ed25519 -a 100 -C "user@eom.ust.edu.ua" -f
↪  ~/user_ed25519
```

В результаті отримуємо 4 файли:

— user_ed25519. Приватний ключ ED25519;
— user_ed25519.pub. Публічний ключ ED25519;

— user_rsa. Приватний ключ RSA;
— user_rsa.pub. Приватний ключ RSA.

Н.3. Реєстрація ключів

Для надання можливості автентифікації за ключем додамо публічні ключі у файл authorized_keys. Для цього необхідно створити директорію .ssh у директорії користувача, якщо вона відсутня. Ця директорія та файли в ній мають мати певні права доступу. Інакше служба ssh, в цілях безпеки, буде їх ігнорувати. Для створення директорії та налаштування прав доступу використовуємо наступні команди. Для створення каталогу використовуємо програму mkdir [60]. Права доступу змінюємо з використанням програми chmod [61].

```
[user@hostname /]# mkdir -p ~/.ssh/
[user@hostname /]# chmod 700 ~/.ssh/
```

Створюємо файл authorized_keys і додаємо в нього публічні ключі користувача (наприклад вміст файлу user_ed25519.pub, див. розділ Н.2). Потім надаємо йому необхідні права доступу.

```
[user@hostname /]# nano ~/.ssh/authorized_keys
[user@hostname /]# chmod 644 ~/.ssh/authorized_keys
```

ДОДАТОК П
КОНФІГУРАЦІЙНИЙ ФАЙЛ NTPD

```
# Please consider joining the pool:
#
#     http://www.pool.ntp.org/join.html
#
# For additional information see:
# - https://wiki.archlinux.org/index.php/Network_Time_Protocol_daemon
# - http://support.ntp.org/bin/view/Support/GettingStarted
# - the ntp.conf man page

# Ukraine pool
pool 0.ua.pool.ntp.org iburst
pool 1.ua.pool.ntp.org iburst
pool 2.ua.pool.ntp.org iburst
pool 3.ua.pool.ntp.org iburst

# World pool
pool 0.pool.ntp.org iburst
pool 1.pool.ntp.org iburst
pool 2.pool.ntp.org iburst
pool 3.pool.ntp.org iburst

tos orphan 15

restrict default kod limited nomodify notrap nopeer noquery
restrict -6 default kod limited nomodify notrap nopeer noquery

restrict 127.0.0.1
restrict -6 ::1

driftfile /var/lib/ntp/ntp.drift
logfile /var/log/ntp.log
```

ДОДАТОК Р
ЗОНА EOM.UST.EDU.UA

```
$ORIGIN eom.ust.edu.ua.
$TTL 3600   ; 1 hour
@           IN  SOA ns1.eom.ust.edu.ua. admin.eom.ust.edu.ua. (
                1          ; serial
                3600       ; refresh (1 hour)
                900        ; retry (15 minutes)
                3600       ; expire (1 hour)
                300        ; minimum (5 minutes)
                )
            IN  NS  ns1.eom.ust.edu.ua.
            IN  MX  10 mail.eom.ust.edu.ua.
            IN  A   10.0.0.1

mail        IN  A   10.0.0.1
ns1         IN  A   10.0.0.1
server      IN  A   10.0.0.1
```

ДОДАТОК С
ЗВОРОТНА ЗОНА EOM.UST.EDU.UA

```
$TTL 3600   ; 1 hour
@                       IN  SOA eom.ust.edu.ua. admin.eom.ust.edu.ua. (
                            1          ; serial
                            3600       ; refresh (1 hour)
                            900        ; retry (15 minutes)
                            3600       ; expire (1 hour)
                            300        ; minimum (5 minutes)
                            )
                        IN  NS  eom.ust.edu.ua.

1.0.0.10.in-addr.arpa.  IN  PTR eom.ust.edu.ua.
1.0.0.10.in-addr.arpa.  IN  PTR mail.eom.ust.edu.ua.
1.0.0.10.in-addr.arpa.  IN  PTR ns1.eom.ust.edu.ua.
1.0.0.10.in-addr.arpa.  IN  PTR server.eom.ust.edu.ua.
```

ДОДАТОК Т
ЗОНА LXC

```
$ORIGIN lxc.;
$TTL 3600   ; 1 hour
@           IN  SOA ns1.lxc. admin.lxc.(
                1          ; serial
                3600       ; refresh (1 hour)
                900        ; retry (15 minutes)
                3600       ; expire (1 hour)
                300        ; minimum (5 minutes)
                )
            IN  NS  ns1.lxc.
            IN  MX  10  mail.lxc.
            IN  A   10.0.3.1

ns1         IN  A   10.0.3.1
mail        IN  A   10.0.3.1
```

ДОДАТОК У
ЗВОРОТНА ЗОНА LXC

```
$TTL 3600   ; 1 hour
@                       IN  SOA lxc. admin.lxc. (
                            1          ; serial
                            3600       ; refresh (1 hour)
                            900        ; retry (15 minutes)
                            3600       ; expire (1 hour)
                            300        ; minimum (5 minutes)
                            )
                        IN  NS  lxc.

1.3.0.10.in-addr.arpa.  IN  PTR lxc.
1.3.0.10.in-addr.arpa.  IN  PTR mail.lxc.
1.3.0.10.in-addr.arpa.  IN  PTR ns1.lxc.
```

ДОДАТОК Ф
ВІДОМОСТІ ПРО КОРЕНЕВІ СЕРВЕРИ

```
;       This file holds the information on root name servers needed to
;       initialize cache of Internet domain name servers
;       (e.g. reference this file in the "cache  .  <file>"
;       configuration file of BIND domain name servers).
;
;       This file is made available by InterNIC
;       under anonymous FTP as
;           file                /domain/named.cache
;           on server           FTP.INTERNIC.NET
;       -OR-                    RS.INTERNIC.NET
;
;       last update:     April 27, 2023
;       related version of root zone:     2023042701
;
; FORMERLY NS.INTERNIC.NET
;
.                        3600000      NS    A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.      3600000      A     198.41.0.4
A.ROOT-SERVERS.NET.      3600000      AAAA  2001:503:ba3e::2:30
;
; FORMERLY NS1.ISI.EDU
;
.                        3600000      NS    B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET.      3600000      A     199.9.14.201
B.ROOT-SERVERS.NET.      3600000      AAAA  2001:500:200::b
;
; FORMERLY C.PSI.NET
;
.                        3600000      NS    C.ROOT-SERVERS.NET.
C.ROOT-SERVERS.NET.      3600000      A     192.33.4.12
C.ROOT-SERVERS.NET.      3600000      AAAA  2001:500:2::c
;
; FORMERLY TERP.UMD.EDU
;
.                        3600000      NS    D.ROOT-SERVERS.NET.
D.ROOT-SERVERS.NET.      3600000      A     199.7.91.13
D.ROOT-SERVERS.NET.      3600000      AAAA  2001:500:2d::d
;
; FORMERLY NS.NASA.GOV
;
.                        3600000      NS    E.ROOT-SERVERS.NET.
E.ROOT-SERVERS.NET.      3600000      A     192.203.230.10
E.ROOT-SERVERS.NET.      3600000      AAAA  2001:500:a8::e
;
```

```
; FORMERLY NS.ISC.ORG
;
.                       3600000      NS    F.ROOT-SERVERS.NET.
F.ROOT-SERVERS.NET.     3600000      A     192.5.5.241
F.ROOT-SERVERS.NET.     3600000      AAAA  2001:500:2f::f
;
; FORMERLY NS.NIC.DDN.MIL
;
.                       3600000      NS    G.ROOT-SERVERS.NET.
G.ROOT-SERVERS.NET.     3600000      A     192.112.36.4
G.ROOT-SERVERS.NET.     3600000      AAAA  2001:500:12::d0d
;
; FORMERLY AOS.ARL.ARMY.MIL
;
.                       3600000      NS    H.ROOT-SERVERS.NET.
H.ROOT-SERVERS.NET.     3600000      A     198.97.190.53
H.ROOT-SERVERS.NET.     3600000      AAAA  2001:500:1::53
;
; FORMERLY NIC.NORDU.NET
;
.                       3600000      NS    I.ROOT-SERVERS.NET.
I.ROOT-SERVERS.NET.     3600000      A     192.36.148.17
I.ROOT-SERVERS.NET.     3600000      AAAA  2001:7fe::53
;
; OPERATED BY VERISIGN, INC.
;
.                       3600000      NS    J.ROOT-SERVERS.NET.
J.ROOT-SERVERS.NET.     3600000      A     192.58.128.30
J.ROOT-SERVERS.NET.     3600000      AAAA  2001:503:c27::2:30
;
; OPERATED BY RIPE NCC
;
.                       3600000      NS    K.ROOT-SERVERS.NET.
K.ROOT-SERVERS.NET.     3600000      A     193.0.14.129
K.ROOT-SERVERS.NET.     3600000      AAAA  2001:7fd::1
;
; OPERATED BY ICANN
;
.                       3600000      NS    L.ROOT-SERVERS.NET.
L.ROOT-SERVERS.NET.     3600000      A     199.7.83.42
L.ROOT-SERVERS.NET.     3600000      AAAA  2001:500:9f::42
;
; OPERATED BY WIDE
;
.                       3600000      NS    M.ROOT-SERVERS.NET.
M.ROOT-SERVERS.NET.     3600000      A     202.12.27.33
M.ROOT-SERVERS.NET.     3600000      AAAA  2001:dc3::35
; End of file
```

ДОДАТОК Х
ЗОНА LOCALHOST

```
@           1D  IN  SOA localhost. root.localhost. (
                    42  ; serial (yyyymmdd##)
                    3H  ; refresh
                    15M ; retry
                    1W  ; expiry
                    1D  ; minimum ttl
                    )

            1D  IN  NS      localhost.

localhost.  1D  IN  A       127.0.0.1
localhost.  1D  IN  AAAA    ::1
```

ДОДАТОК Ц
ЗВОРОТНА ЗОНА LOCAHOST

```
@                          1D  IN  SOA localhost. root.localhost. (
                                       42  ; serial (yyyymmdd##)
                                       3H  ; refresh
                                       15M ; retry
                                       1W  ; expiry
                                       1D  ; minimum ttl
                                       )

                           1D  IN  NS      localhost.

1.0.0.127.in-addr.arpa. 1D  IN  PTR     localhost.
```

ДОДАТОК Ш
ЗВОРОТНА ЗОНА LOCAHOST IPV6

```
@       1D  IN  SOA   localhost. root.localhost. (
                42    ; serial (yyyymmdd##)
                3H    ; refresh
                15M   ; retry
                1W    ; expiry
                1D )  ; minimum ttl

        1D  IN  NS      localhost.

1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa.   1D  IN
↪  PTR localhost.
```

ДОДАТОК Щ
КОНФІГУРАЦІЙНИЙ ФАЙЛ NAMED

```
// vim:set ts=4 sw=4 et:

acl local {
    127.0.0.1;
    ::1;
};

options {
    directory "/var/named";
    pid-file "/run/named/named.pid";

    // Увімкнути підтримку IPv4 та IPv6
    //listen-on-v6 { any; };
    listen-on { any; };


    // Дозволити обробку запитів
    allow-query { any; };
    allow-recursion { any; };

    // Заборонити передачу зони
    allow-transfer { none; };

    // Дозволити оновлення лише з локального з'єднання
    allow-update { local; };


    version none;
    hostname none;
    server-id none;

    empty-zones-enable yes;
    recursion yes;
};

logging {
    channel xfer-log {
        file "/var/log/named.log";
            print-category yes;
            print-severity yes;
            severity info;
        };
        category xfer-in { xfer-log; };
        category xfer-out { xfer-log; };
```

```
        category notify { xfer-log; };
};

include "/var/named/dns.keys.conf";

// zone file for the root servers
// discretionary zone (see root server discussion above)
zone "." {
    type hint;
    file "named.root";
};

zone "localhost" {
    type primary;
    file "localhost.zone";
    notify no;
};

zone "0.0.127.in-addr.arpa" {
    type primary;
    file "localhost.rev";
    notify no;
};

zone "1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa" {
    type primary;
    file "localhost.ip6.rev";
    notify no;
};

zone "eom.ust.edu.ua" {
    type primary;
    file "eom.ust.edu.ua.zone";
    notify yes;
};

zone "0.0.10.in-addr.arpa" {
    type primary;
    file "eom.ust.edu.ua.rev";
    notify yes;
};

zone "lxc" {
    type primary;
    file "lxc.zone";
    notify yes;
};
```

```
zone "3.0.10.in-addr.arpa" {
    type primary;
    file "lxc.rev";
    notify yes;
};
```

ДОДАТОК Ю
КОНФІГУРАЦІЙНИЙ ФАЙЛ KEA-DDNS

```
// This is a basic configuration for the Kea DHCP DDNS daemon.
//
// This is just a very basic configuration. Kea comes with large suite (over 30)
// of configuration examples and extensive Kea User's Guide. Please refer to
// those materials to get better understanding of what this software is able to
// do. Comments in this configuration file sometimes refer to sections for more
// details. These are section numbers in Kea User's Guide. The version matching
// your software should come with your Kea package, but it is also available
// in ISC's Knowledgebase (https://kea.readthedocs.io; the direct link for
// the stable version is https://kea.readthedocs.io/).
//
// This configuration file contains only DHCP DDNS daemon's configuration.
// If configurations for other Kea services are also included in this file they
// are ignored by the DHCP DDNS daemon.
{
  // DHCP DDNS configuration starts here. This is a very simple configuration
  // that simply starts the DDNS daemon, but will not do anything useful.
  // See Section 11 for examples and details description.
  "DhcpDdns": {
    "ip-address": "127.0.0.1",
    "port": 53001,
    "control-socket": {
      "socket-type": "unix",
      "socket-name": "/tmp/kea-ddns-ctrl-socket"
    },
    "tsig-keys": [
      {
        "name":"localhost",
        "algorithm":"HMAC-SHA256",
        "secret":"{CEKPET}"
      }
    ],
    "forward-ddns": {
      "ddns-domains": [
        {
          "name": "eom.ust.edu.ua.",
          "key-name": "localhost",
          "dns-servers": [
            {
              "ip-address": "127.0.0.1"
            }
          ]
        },
        {
```

```
      "name": "lxc.",
      "key-name": "localhost",
      "dns-servers": [
        {
          "ip-address": "127.0.0.1"
        }
      ]
    }
  ]
},
"reverse-ddns": {
  "ddns-domains": [
    {
      "name": "0.0.10.in-addr.arpa.",
      "key-name": "localhost",
      "dns-servers": [
        {
          "ip-address": "127.0.0.1"
        }
      ]
    },
    {
      "name": "3.0.10.in-addr.arpa.",
      "key-name": "localhost",
      "dns-servers": [
        {
          "ip-address": "127.0.0.1"
        }
      ]
    }
  ]
},
// Logging configuration starts here. Kea uses different loggers to log various
// activities. For details (e.g. names of loggers), see Chapter 18.
"loggers": [
  {
    // This specifies the logging for D2 (DHCP-DDNS) daemon.
    "name": "kea-dhcp-ddns",
    "output_options": [
      {
        // Specifies the output file. There are several special values
        // supported:
        // - stdout (prints on standard output)
        // - stderr (prints on standard error)
        // - syslog (logs to syslog)
        // - syslog:name (logs to syslog using specified name)
        // Any other value is considered a name of the file
        // "output": "/var/log/kea-ddns.log"
```

```
            "output": "stdout"
            // Shorter log pattern suitable for use with systemd,
            // avoids redundant information
            // "pattern": "%-5p %m\n"
            // This governs whether the log output is flushed to disk after
            // every write.
            // "flush": false,
            // This specifies the maximum size of the file before it is
            // rotated.
            // "maxsize": 1048576,
            // This specifies the maximum number of rotated files to keep.
            // "maxver": 8
          }
        ],
        // This specifies the severity of log messages to keep. Supported values
        // are: FATAL, ERROR, WARN, INFO, DEBUG
        "severity": "INFO",
        // If DEBUG level is specified, this value is used. 0 is least verbose,
        // 99 is most verbose. Be cautious, Kea can generate lots and lots
        // of logs if told to do so.
        "debuglevel": 0
      }
    ]
  }
}
```

ДОДАТОК Я
КОНФІГУРАЦІЙНИЙ ФАЙЛ KEA-DHCP4

```
// This is a basic configuration for the Kea DHCPv4 server. Subnet declarations
// are mostly commented out and no interfaces are listed. Therefore, the servers
// will not listen or respond to any queries.
// The basic configuration must be extended to specify interfaces on which
// the servers should listen. There are a number of example options defined.
// These probably don't make any sense in your network. Make sure you at least
// update the following, before running this example in your network:
// - change the network interface names
// - change the subnets to match your actual network
// - change the option values to match your network
//
// This is just a very basic configuration. Kea comes with large suite (over 30)
// of configuration examples and extensive Kea User's Guide. Please refer to
// those materials to get better understanding of what this software is able to
// do. Comments in this configuration file sometimes refer to sections for more
// details. These are section numbers in Kea User's Guide. The version matching
// your software should come with your Kea package, but it is also available
// in ISC's Knowledgebase (https://kea.readthedocs.io; the direct link for
// the stable version is https://kea.readthedocs.io/).
//
// This configuration file contains only DHCPv4 server's configuration.
// If configurations for other Kea services are also included in this file they
// are ignored by the DHCPv4 server.
{
    // DHCPv4 configuration starts here. This section will be read by DHCPv4 server
    // and will be ignored by other components.
    "Dhcp4": {
        "reservations-global": true,
        "reservations-in-subnet": false,
        // Add names of your network interfaces to listen on.
        "interfaces-config": {
            // Дозволяємо обробку всіх інтерфейсів.
            // Детальніша конфігурація інтерфейсів буде описана в мережах
            "interfaces": [
                "*"
            ]
        },
        "multi-threading": {
            "enable-multi-threading": true,
            "thread-pool-size": 4,
            "packet-queue-size": 28
        },
        // Kea supports control channel, which is a way to receive management
        // commands while the server is running. This is a Unix domain socket that
```

```
// receives commands formatted in JSON, e.g. config-set (which sets new
// configuration), config-reload (which tells Kea to reload its
// configuration from file), statistic-get (to retrieve statistics) and
↪  many
// more. For detailed description, see Sections 8.8, 16 and 15.
"control-socket": {
    "socket-type": "unix",
    "socket-name": "/tmp/kea4-ctrl-socket"
},
// Use Memfile lease database backend to store leases in a CSV file.
// Depending on how Kea was compiled, it may also support SQL databases
// (MySQL and/or PostgreSQL). Those database backends require more
// parameters, like name, host and possibly user and password.
// There are dedicated examples for each backend. See Section 7.2.2 "Lease
// Storage" for details.
"lease-database": {
   // Memfile is the simplest and easiest backend to use. It's an in-memory
    // C++ database that stores its state in CSV file.
    "type": "memfile",
    "lfc-interval": 3600
},
// See Section 7.2.3 "Hosts storage" for details.
// Setup reclamation of the expired leases and leases affinity.
// Expired leases will be reclaimed every 10 seconds. Every 25
// seconds reclaimed leases, which have expired more than 3600
// seconds ago, will be removed. The limits for leases reclamation
// are 100 leases or 250 ms for a single cycle. A warning message
// will be logged if there are still expired leases in the
// database after 5 consecutive reclamation cycles.
"expired-leases-processing": {
    "reclaim-timer-wait-time": 10,
    "flush-reclaimed-timer-wait-time": 25,
    "hold-reclaimed-time": 3600,
    "max-reclaim-leases": 100,
    "max-reclaim-time": 250,
    "unwarned-reclaim-cycles": 5
},
// Global timers specified here apply to all subnets, unless there are
// subnet specific values defined in particular subnets.
"renew-timer": 900,
"rebind-timer": 1800,
"valid-lifetime": 3600,
// Below an example of a simple IPv4 subnet declaration. Uncomment to enable
// it. This is a list, denoted with [ ], of structures, each denoted with
// { }. Each structure describes a single subnet and may have several
// parameters. One of those parameters is "pools" that is also a list of
// structures.
"subnet4": [
```

```
{
    // Назва домену мережі
    "ddns-qualifying-suffix": "eom.ust.edu.ua.",
    "ddns-send-updates": true,
    // Інтерфейс якому відповідає поточна мережа
    "interface": "eno2",
    // This defines the whole subnet. Kea will use this information to
    // determine where the clients are connected. This is the whole
    // subnet in your network. This is mandatory parameter for each
    // subnet.
    "subnet": "10.0.0.0/24",
    // Pools define the actual part of your subnet that is governed
    // by Kea. Technically this is optional parameter, but it's
    // almost always needed for DHCP to do its job. If you omit it,
    // clients won't be able to get addresses, unless there are
    // host reservations defined for them.
    "pools": [
        {
            "pool": "10.0.0.25 - 10.0.0.254"
        }
    ],
    // These are options that are subnet specific. In most cases,
    // you need to define at least routers option, as without this
    // option your clients will not be able to reach their default
    // gateway and will not have Internet connectivity.
    "option-data": [
        {
            // Адреса шлюзу
            "name": "routers",
            "data": "10.0.0.1"
        },
        {
            // Адреса серверу часу
            "name": "ntp-servers",
            "data": "10.0.0.1"
        },
        {
            // Адреса DNS
            "name": "domain-name-servers",
            "data": "10.0.0.1"
        },
        {
            // Домен підстанови
            "name": "domain-search",
            "data": "eom.ust.edu.ua."
        },
        {
            "name": "dhcp-server-identifier",
```

```
                "data": "10.0.0.1"
            },
            {
                "name": "domain-name",
                "data": "eom.ust.edu.ua."
            }
        ],
       // Kea offers host reservations mechanism. Kea supports reservations
        // by several different types of identifiers: hw-address
       // (hardware/MAC address of the client), duid (DUID inserted by the
       // client), client-id (client identifier inserted by the client) and
        // circuit-id (circuit identifier inserted by the relay agent).
        //
        // Kea also support flexible identifier (flex-id), which lets you
       // specify an expression that is evaluated for each incoming packet.
        // Resulting value is then used for as an identifier.
        //
        // Note that reservations are subnet-specific in Kea. This is
        // different than ISC DHCP. Keep that in mind when migrating
        // your configurations.
        "reservations": []
    },
    {
        // Назва домену мережі
        "ddns-qualifying-suffix": "lxc.",
        "ddns-send-updates": true,
        // Інтерфейс якому відповідає поточна мережа
        "interface": "lxcbr0",
        // This defines the whole subnet. Kea will use this information to
        // determine where the clients are connected. This is the whole
        // subnet in your network. This is mandatory parameter for each
        // subnet.
        "subnet": "10.0.3.0/24",
        // Pools define the actual part of your subnet that is governed
        // by Kea. Technically this is optional parameter, but it's
        // almost always needed for DHCP to do its job. If you omit it,
        // clients won't be able to get addresses, unless there are
        // host reservations defined for them.
        "pools": [
            {
                "pool": "10.0.3.25 - 10.0.3.254"
            }
        ],
        // These are options that are subnet specific. In most cases,
        // you need to define at least routers option, as without this
        // option your clients will not be able to reach their default
        // gateway and will not have Internet connectivity.
        "option-data": [
```

```
                {
                    // Адреса шлюзу
                    "name": "routers",
                    "data": "10.0.3.1"
                },
                {
                    // Адреса серверу часу
                    "name": "ntp-servers",
                    "data": "10.0.3.1"
                },
                {
                    // Адреса DNS
                    "name": "domain-name-servers",
                    "data": "10.0.3.1"
                },
                {
                    // Домен підстановки
                    "name": "domain-search",
                    "data": "lxc."
                },
                {
                    "name": "dhcp-server-identifier",
                    "data": "10.0.3.1"
                },
                {
                    "name": "domain-name",
                    "data": "lxc."
                }
            ],
            // Kea offers host reservations mechanism. Kea supports reservations
            // by several different types of identifiers: hw-address
            // (hardware/MAC address of the client), duid (DUID inserted by the
            // client), client-id (client identifier inserted by the client) and
            // circuit-id (circuit identifier inserted by the relay agent).
            //
            // Kea also support flexible identifier (flex-id), which lets you
            // specify an expression that is evaluated for each incoming packet.
            // Resulting value is then used for as an identifier.
            //
            // Note that reservations are subnet-specific in Kea. This is
            // different than ISC DHCP. Keep that in mind when migrating
            // your configurations.
            "reservations": []
        }
    ],
    // There are many, many more parameters that DHCPv4 server is able to use.
    // They were not added here to not overwhelm people with too much
    // information at once.
```

```
// Logging configuration starts here. Kea uses different loggers to log
↪  various
// activities. For details (e.g. names of loggers), see Chapter 18.
"loggers": [
    {
        // This section affects kea-dhcp4, which is the base logger for
        ↪  DHCPv4
        // component. It tells DHCPv4 server to write all log messages (on
        // severity INFO or more) to a file.
        "name": "kea-dhcp4",
        "output_options": [
            {
                // Specifies the output file. There are several special
                ↪  values
                // supported:
                // - stdout (prints on standard output)
                // - stderr (prints on standard error)
                // - syslog (logs to syslog)
                // - syslog:name (logs to syslog using specified name)
                // Any other value is considered a name of the file
                // "output": "/var/log/kea-dhcp4.log"
                "output": "stdout"
                // Shorter log pattern suitable for use with systemd,
                // avoids redundant information
                // "pattern": "%-5p %m\n",
                // This governs whether the log output is flushed to disk
                ↪  after
                // every write.
                // "flush": false,
               // This specifies the maximum size of the file before it is
                // rotated.
                // "maxsize": 1048576,
                // This specifies the maximum number of rotated files to
                ↪  keep.
                // "maxver": 8
            }
        ],
        // This specifies the severity of log messages to keep. Supported
        ↪  values
        // are: FATAL, ERROR, WARN, INFO, DEBUG
        "severity": "INFO",
        // If DEBUG level is specified, this value is used. 0 is least
        ↪  verbose,
        // 99 is most verbose. Be cautious, Kea can generate lots and lots
        // of logs if told to do so.
        "debuglevel": 0
    }
],
```

```
        // DDNS information (how the DHCPv6 component can reach a DDNS daemon)
        "ddns-generated-prefix": "host",
        "ddns-replace-client-name": "when-not-present",
        "ddns-update-on-renew": true,
        "ddns-override-client-update": true,
        "dhcp-ddns": {
            "enable-updates": true,
            "server-ip": "127.0.0.1",
            "ncr-protocol": "UDP",
            "ncr-format": "JSON"
        }
    }
}
```

ДОДАТОК АА
СЕРВЕРНИЙ НАБІР ПРАВИЛ БРАНДМАУЕРА

```
# Static information about the filesystems.
# See fstab(5) for details.

# <file system> <dir> <type> <options> <dump> <pass>
UUID=be48508f-45a1-462c-85d8-368c7086a657       /       ext4    defaults
 ↪  0       1
UUID=0879-0B5E                                  /efi    vfat
 ↪  defaults,fat=32,umask=0077      0       2
UUID=cd4b8675-362e-4874-8dec-ea7692946e27       /var    ext4    defaults
 ↪  0       2
UUID=6fc4811b-7cb2-48ab-81c7-fe95284fb9e8       none    swap    defaults
 ↪  0       0
tmpfs                                           /tmp    tmpfs
 ↪  defaults,size=8G,mode=1777      0       0
```