

Міністерство освіти і науки України

Луцький національний технічний університет

(повне найменування закладу вищої освіти)

Факультет комп'ютерних та інформаційних технологій

(повне найменування факультету)

Кафедра комп'ютерної інженерії та безпеки

(повне найменування кафедри)

КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «МАГІСТР»

ДОСЛІДЖЕННЯ РІШЕНЬ ТА МОДЕРНІЗАЦІЯ ДОМАШНЬОЇ
СЕРВЕРНОЇ ІНФРАСТРУКТУРИ
RESEARCHING SOLUTIONS AND MODERNIZING HOME
SERVER INFRASTRUCTURE

спеціальність 123 Комп'ютерна інженерія

(шифр і назва спеціальності)

освітня програма Комп'ютерна інженерія

(назва освітньої програми)

Виконав: здобувач вищої освіти
групи КІМ-21
Слюсар Максим Ярославович

(підпис)

Керівник:
к.т.н., доцент
Багнюк Наталія Володимирівна

(підпис)

Кваліфікаційну роботу
допущено до захисту
« _____ » _____ грудня _____ 2025 р.

Гарант освітньої програми:
к.т.н., доцент
Гринюк Сергій Васильович

(підпис)

Луцьк – 2025 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра комп'ютерної інженерії та безпеки

Ступінь вищої освіти: магістр

Галузь знань: 12 Інформаційні технології

Спеціальність: 123 Комп'ютерна інженерія

Освітня програма: «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

доц. Т.ТЕРЛЕЦЬКИЙ

« _____ » _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Слюсару Максиму Ярославовичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи Дослідження рішень та модернізація домашньої серверної інфраструктури

Керівник роботи к.т.н., доцент Багнюк Наталія Володимирівна

затверджені наказом закладу вищої освіти від «17» червня 2025 року № 290/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи 09.12.2025р.

3. Вихідні дані до роботи Джерелом розробки є науково-технічна література та публікації в періодичних виданнях з даного питання, опубліковані зарубіжні та вітчизняні роботи в даній області, різні інтернет-ресурси технічного спрямування

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

Вступ

Основні концепції домашнього сервера

Аналіз та проектування домашньої серверної інфраструктури

Дослідження ефективності домашньої серверної інфраструктури

Висновки

5. Перелік графічного (ілюстративного) матеріалу:

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
<i>Основні концепції домашнього сервера</i>	<i>Багнюк Н.В., доцент</i>		
<i>Аналіз та проектування домашньої серверної інфраструктури</i>	<i>Багнюк Н.В., доцент</i>		
<i>Дослідження ефективності домашньої серверної інфраструктури</i>	<i>Багнюк Н.В., доцент</i>		
<i>Нормоконтроль</i>	<i>Багнюк Н.В., доцент</i>		
<i>Гарант ОП</i>	<i>Гринюк С.В., доцент</i>		
<i>Показник запозичень тексту</i>		%	
<i>Академічна доброчесність</i>	<i>Міскевич О.І., ст.викладач</i>		

7. Дата видачі завдання

18.06.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	<i>Огляд літератури із досліджуваної проблеми</i>	До 01.08.2025 р.	
2.	<i>Аналіз та проектування домашньої серверної інфраструктури та дослідження ефективності домашньої серверної інфраструктури</i>	До 20.08.2025 р.	
3.	<i>Дослідження ефективності домашньої серверної інфраструктури</i>	До 25.09.2025 р.	
4.	<i>Практична реалізація об'єкта проектування</i>	До 20.10.2025 р.	
5.	<i>Висновки та пропозиції</i>	До 25.10.2025 р.	
6.	<i>Формування списку використаних джерел</i>	До 27.10.2025 р.	
7.	<i>Формування додатків</i>	До 30.10.2025 р.	
8.	<i>Оформлення ілюстративного матеріалу</i>	До 05.11.2025 р.	
9.	<i>Представлення остаточного варіанту кваліфікаційної роботи керівникові</i>	До 11.11.2025 р.	
10.	<i>Нормоконтроль</i>	До 29.11.2025 р.	
11.	<i>Інструментальна перевірка на академічний плагіат</i>	До 02.12.2025 р.	
12.	<i>Здача кваліфікаційної роботи та всіх супровідних документів на кафедрі</i>	До 09.12.2025 р.	

Здобувач вищої освіти

(підпис)

Слюсар М.Я.

(прізвище, ініціали)

Керівник кваліфікаційної роботи

(підпис)

Багнюк Н.В.

(прізвище, ініціали)

АНОТАЦІЯ

Слюсар М.Я. Дослідження рішень та модернізація домашньої серверної інфраструктури.

Кваліфікаційна робота магістра ОП «Комп'ютерна інженерія» спеціальності 123 Комп'ютерна інженерія. Луцький національний технічний університет. Луцьк, 2025.

Кваліфікаційна робота складається з вступу, трьох розділів, висновків, списку використаних джерел, додатків.

Перший розділ присвячено огляду предметної області, тут розглядаються основні поняття про сервери, їх види та сфери їх використання, наведено практичні приклади. Також в цьому розділі здійснено огляд рішень для безпечного користування.

В другому розділі здійснено вибір засобів інфраструктури. Обрано операційні системи. Здійснено порівняння можливостей операційних систем.

Третій розділ присвячено опису дослідження та порівнянню можливостей конфігурацій. Описано процес обслуговування пристрою та модернізації апаратної частини. Здійснено первинне налаштування систем, та проведено заходи для підвищення ефективності. Окрім цього продемонстровано процес налаштування служб. Проведено тестування. Описано висновки згідно з результатами дослідження.

Ключові слова: сервер, інфраструктура, медіастрімінг, доступ, операційна система, файлова система, сховище, конфігурація, SSD, HDD.

ANNOTATION

Sliusar M. Researching solutions and modernizing home server infrastructure.

Qualifying work of a Master's of EP «Computer Engineering» specialty 123 Computer Engineering. Lutsk National Technical University. Lutsk, 2025.

Qualification work consists of an introduction, three sections, conclusions, a reference, three appendices.

The first section is devoted to an overview of the subject area, here the basic concepts of servers, their types and areas of their use are considered, practical examples are given. This section also reviews solutions for secure use.

In the second section, the infrastructure tools are selected. Operating systems are selected. The capabilities of operating systems are compared.

The third section is devoted to a description of the study and a comparison of the capabilities of configurations. The process of device maintenance and hardware modernization is described. The initial system setup is carried out, and daily efficiency improvement measures are carried out. In addition, the process of service setup is demonstrated. Testing is carried out. The conclusions according to the results of the study are described.

Keywords: server, infrastructure, media streaming, access, operating system, file system, storage, configuration, SSD, HDD.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1 ОСНОВНІ КОНЦЕПЦІЇ ДОМАШНЬОГО СЕРВЕРА	9
1.1 Поняття та призначення домашнього сервера	9
1.2 Апаратні компоненти.....	11
1.3 Програмні рішення. Огляд операційних систем.....	14
1.4 Функціональні можливості домашніх серверів	17
1.5 Безпека та моніторинг	22
1.6 Аналіз готових рішень.....	25
РОЗДІЛ 2 АНАЛІЗ ТА ПРОЕКТУВАННЯ ДОМАШНЬОЇ СЕРВЕРНОЇ ІНФРАСТРУКТУРИ	30
2.1 Моделі побудови серверної інфраструктури	30
2.2 Критерії вибору операційної системи.....	34
2.3 Порівняння можливостей ОС	37
2.4 Критерії та вибір програмного забезпечення.....	40
РОЗДІЛ 3 ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ДОМАШНЬОЇ СЕРВЕРНОЇ ІНФРАСТРУКТУРИ	43
3.1 Планування дослідження	43
3.2 Підготовка пристрою	45
3.3 Первинне налаштування конфігурацій.....	48
3.4 Модернізація ОС	52
3.5 Встановлення та налаштування сервісів	60
3.6 Тестування та оцінка результатів	69
3.7 Висновки з реалізації	72
ВИСНОВКИ.....	73
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	75
ДОДАТКИ.....	78

ВСТУП

У сучасному світі розвиток цифрових технологій та зростання кількості пристроїв призводить до підвищення вимог зберігання, обробки та передачі даних. Все більше користувачів прагнуть створювати власні локальні рішення для зберігання інформації, розгортання веб-сервісів або власних кінотеатрів. Одним з найефективніших рішень для таких завдань є створення домашньої серверної інфраструктури.

Домашні сервери дозволяють користувачу отримати незалежність від комерційних хмарних сервісів, забезпечити конфіденційність даних, налаштувати власні сервіси для роботи з файлами, медіа або мережею. Навіть попри доступність недорогого обладнання та відкритого програмного забезпечення, побудова і підтримка такої інфраструктури вимагає технічних знань і розуміння принципів роботи серверних систем. Також актуальним залишається питання модернізації таких рішень з метою підвищення їх надійності, енергоефективності та продуктивності.

Актуальність теми. Актуальність дослідження полягає в зростаючій потребі в автономних гнучких та безпечних рішеннях для зберігання даних і роботи сервісів у домашніх умовах. Модернізація домашньої серверної інфраструктури дозволяє оптимізувати використання ресурсів, підвищити ефективність роботи операційних систем і забезпечити стабільну роботу навіть при обмежених апаратних можливостях.

Метою роботи є дослідження, аналіз і практична реалізація рішень щодо модернізації домашньої серверної інфраструктури, які будуть спрямовані на покращення її продуктивності стабільності та енергоефективності.

Об'єкт дослідження – процес побудови та модернізації домашньої серверної інфраструктури.

Предмет дослідження – апаратні та програмні рішення, що застосовуються для підвищення ефективності роботи домашніх серверів.

Завдання дослідження:

- провести аналіз існуючих рішень і підходів до організації домашніх серверів;

- вибір апаратного та програмного забезпечення для побудови серверної інфраструктури;

- реалізація процесу оптимізації та модернізації обраних систем;

- оцінити результати здійснених операцій та визначити їх ефективність.

Апробація результатів. Результати роботи опубліковані в науковому журналі «Комп'ютерно-інтегровані технології: освіта, наука, виробництво», м. Луцьк, 2025. № 60. С. 70-77 [1].

РОЗДІЛ 1

ОСНОВНІ КОНЦЕПЦІЇ ДОМАШНЬОГО СЕРВЕРА

1.1 Поняття та призначення домашнього сервера

Домашній сервер – це пристрій, який виконує функції зберігання обробки та передачі даних у межах домашньої або невеликої локальної мережі. На відміну від звичайного персонального комп'ютера, який призначений для індивідуального використання, сервер виконує роль центрального вузла, що забезпечує доступ до спільних ресурсів для кількох користувачів і пристроїв одночасно. Головною метою створення домашнього сервера є оптимізація взаємодії між пристроями в межах однієї мережі, забезпечення централізованого зберігання даних, резервного копіювання, потокової передачі медіа, а також надання інших мережевих сервісів.

Поняття домашній сервер з'явилося разом із розвитком цифрових технологій та зростанням кількості пристроїв, що підключаються до Інтернету. Раніше подібна інфраструктура зустрічалась переважно в корпоративних середовищах. Сьогодні завдяки здешевленню апаратного забезпечення, появи компактних одноплатних комп'ютерів і простоті налаштування серверного програмного забезпечення створення власного домашнього сервера стало доступним практично кожному користувачу. Домашній сервер може функціонувати як постійно активна система, яка забезпечує безперервну роботу сервісів різного призначення [2].

Призначення домашнього сервера визначається його функціональними можливостями. Найпоширенішим є використання сервера як центра зберігання даних. У цьому випадку він виконує роль файлового сервера, на якому зберігаються документи, фотографії, відео, резервні копії та інші файли. Завдяки підтримці мережевих протоколів користувачі можуть отримувати доступ до цих даних з інших пристроїв, незалежно від операційної системи. Це забезпечує зручність спільного користування файлами та підвищує рівень

безпеки інформації оскільки всі дані зберігаються локально без необхідності передавати їх на сторонні хмарні платформи [3].

Іншою важливою функцією домашнього сервера є організація мультимедійного центру. За допомогою програм сервер може транслювати відео, музику чи фотографії на інші пристрої через локальну мережу. Це дозволяє створити єдину систему для зберігання та відтворення контенту без потреби дублювати файли на різних пристроях. Медіасервер також може автоматично індексувати медіафайли, завантажувати метадані і обкладинки та забезпечувати потокове відтворення навіть поза межами локальної мережі.

Окрім цього домашній сервер часто виконує роль системи резервного копіювання. У разі втрати або пошкодження даних на основному пристрої користувач може швидко відновити інформацію з копії, що зберігається на сервері. Це особливо актуально в умовах зростаючих ризиків кіберзагроз, збоїв у роботі або помилок користувачів. Використання автоматизованих рішень для резервного копіювання дозволяє підтримувати актуальний стан даних без необхідності втручання користувача.

Серед додаткових можливостей можна виділити веб-хостинг. Розгортання власних сайтів або блогів на локальному сервері, поштовий сервер, систем відеоспостереження з підтримкою IP-камер або контейнеризацію сервісів через Docker. Це робить домашній сервер універсальною платформою, на базі якої можна розгортати будь які сервіси залежно від потреб користувача.

Ще один спосіб використання домашнього сервера, як централізована платформа для керування розумним будинком. Завдяки інтеграції різних систем інтернету речей на одному домашньому сервері можна зручно та легко керувати усіма функціями пристроїв

Важливим аспектом є енергоефективність та безперервна робота. Оскільки сервер зазвичай працює цілодобово, доцільно використовувати пристрої з низьким енергоспоживанням. Використання спеціалізованих пристроїв для серверних функцій дозволяє суттєво знизити витрати на

електроенергію, зберігаючи при цьому високу продуктивність для обраних завдань.

Домашній сервер – це не просто засіб для зберігання даних, а багатофункціональний елемент сучасної цифрової інфраструктури, який забезпечує зручність безпеку автономність і контроль над інформацією в межах локальної мережі. Його впровадження дозволяє користувачу отримати глибше розуміння роботи мережевих технологій, підвищити рівень цифрової грамотності та створити власну інтелектуальну екосистему для зберігання, обміну й захисту даних.

1.2 Апаратні компоненти

Апаратні компоненти домашнього сервера є основною складовою його функціонування. Саме вони визначають рівень продуктивності енергоефективності надійності та масштабованості системи. Незалежно від того, чи сервер створюється на базі старого персонального комп'ютера, спеціалізованого NAS-пристрою або компактного одноплатного комп'ютера на зразок Raspberry Pi (рис. 1.1), правильний вибір і взаємодія апаратних складових безпосередньо впливають на стабільність і ефективність роботи всієї системи.

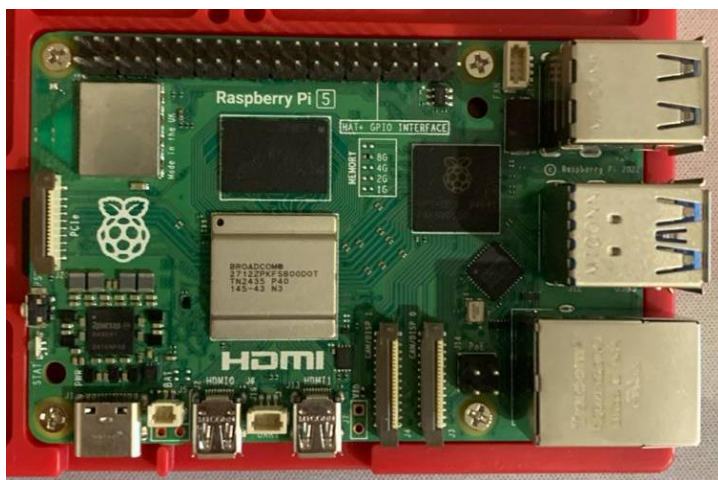


Рисунок 1.1 – Одноплатний комп'ютер Raspberry Pi 5

У класичному розумінні сервер є системою, що складається з: центрального процесора (CPU), оперативної пам'яті (RAM), накопичувачів даних (HDD або SSD), материнської плати, блоку живлення, системи охолодження та мережевого інтерфейсу. Кожен із цих компонентів має свою роль і повинен підбиратися з урахуванням конкретних завдань, які сервер буде виконувати.

Центральний процесор (CPU) є ключовим елементом, що відповідає за обробку даних і виконання всіх обчислювальних операцій. Для домашніх серверів зазвичай не потрібні процесори з високою тактовою частотою або великою кількістю ядер, як у промислових дата-центрах, проте важливо забезпечити баланс між продуктивністю та енергоспоживанням. Для завдань з потокової передачі медіа, роботи з базами даних або резервного копіювання цілком достатньо процесорів середнього рівня або навіть енергоефективних процесорів. Якщо ж сервер буде використовуватись для одночасного виконання кількох віртуальних машин або складних обчислень, доцільно буде обирати багатоядерні процесори.

Оперативна пам'ять (RAM) визначає здатність системи працювати з кількома процесами одночасно. Для більшості домашніх серверів достатнім є обсяг від 8 до 16 ГБ оперативної пам'яті. Проте якщо система виконує роль медіацентру або файлового сховища з великою кількістю одночасних підключень то доцільно буде збільшити обсяг до 16 і більше ГБ оперативної пам'яті. Важливою характеристикою є тип пам'яті DDR3, DDR4 або DDR5.

Накопичувачі даних є одним із найважливіших компонентів домашнього сервера. Саме вони зберігають усі файли, медіаконтент та інші дані. Існують два основні типи накопичувачів: твердотільні SSD і жорсткі диски HDD. SSD відзначаються високою швидкістю читання і запису, що забезпечує швидке завантаження системи та додатків. Проте HDD мають значно більшу ємність та нижчу вартість, тому вони зазвичай використовуються для тривалого зберігання великих обсягів даних. Для підвищення надійності зберігання даних використовуються масиви RAID.

Материнська плата є центральним елементом, який об'єднує всі апаратні компоненти між собою. Вона визначає сумісність із процесором, обсягом і типом пам'яті, кількістю портів, а також наявність інтегрованих мережевих і графічних інтерфейсів. Для домашніх серверів зазвичай обирають компактні форм-фактори, такі як Mini-ITX або Micro-ATX, оскільки вони дозволяють зменшити розміри корпусу і споживання енергії. Під час створення домашнього сервера при виборі материнської плати важливо звернути увагу на підтримку функцій енергозбереження.

Блок живлення забезпечує стабільну подачу електроенергії на всі компоненти системи. Потужність блоку живлення має бути достатньою для живлення всіх компонентів пристрою з певним запасом. Окрему увагу слід приділити якості охолодження блока живлення, оскільки його перегрів може вплинути на стабільність усієї системи.

Система охолодження є ще одним з критично важливих аспектів, оскільки сервер часто працює цілодобово. Для забезпечення безперервної роботи необхідно використовувати ефективні повітряні або рідинні системи охолодження. У домашніх умовах доцільно зупинитися на малошумних вентиляторах і добре продуманій циркуляції повітря в корпусі. Для невеликих одноплатних серверів, таких як Raspberry Pi, достатньо пасивного охолодження у вигляді радіаторів або компактних вентиляторів.

Мережевий інтерфейс забезпечує зв'язок сервера з іншими пристроями локальної мережі. Більшість сучасних материнських плат мають вбудований гігабітний Ethernet-адаптер, який забезпечує швидкість передачі даних до 1 Гбіт/с. У разі бездротового підключення застосовуються Wi-Fi адаптери стандартів 802.11ac або 802.11ax, однак для стабільної та швидкої роботи серверних сервісів перевага завжди надається дротовим з'єднанням[4].

Для енергоефективних і компактних рішень часто використовуються одноплатні комп'ютери. Вони споживають мінімум електроенергії, не потребують складної системи охолодження та забезпечують достатню продуктивність для виконання базових домашніх серверних функцій. У таких

пристроях всі основні компоненти вже інтегровані на одній платі, що робить їх доступними і простими у налаштуванні.

1.3 Програмні рішення. Огляд операційних систем

В процесі створення та експлуатації домашнього сервера вибір програмного забезпечення є одним із ключових чинників, що визначають стабільність безпеку та функціональність усієї системи. Саме операційна система забезпечує взаємодію апаратної частини з програмними сервісами, керує ресурсами пристрою, регулює доступ користувачів та відповідає за безперервну роботу мережевих служб. Для домашніх серверів можуть використовуватися різні типи операційних систем. В ролі операційної системи може виступати як універсальні дистрибутиви так і спеціалізовані серверні платформ або навіть спрощені рішення для одноплатних комп'ютерів.

Основне завдання програмної частини домашнього сервера забезпечити стабільну роботу необхідних сервісів при мінімальних затратах ресурсів. Від вибору операційної системи залежить сумісність з обраним обладнанням, зручність адміністрування, безпека та можливості масштабування. Зазвичай розглядаються три основні напрями: дистрибутиви Linux, системи на базі Windows, а також легкі ОС для одноплатних платформ.

1.3.1 Операційні системи сімейства Linux

Linux – найпоширеніша операційна система для домашніх серверів завдяки своїй гнучкості, стабільності та відкритому вихідному коду. Дистрибутиви Linux надають користувачам повний контроль над системою, можливість налаштовувати будь які параметри та встановлювати лише ті компоненти, які необхідні для конкретного завдання (рис. 1.2).

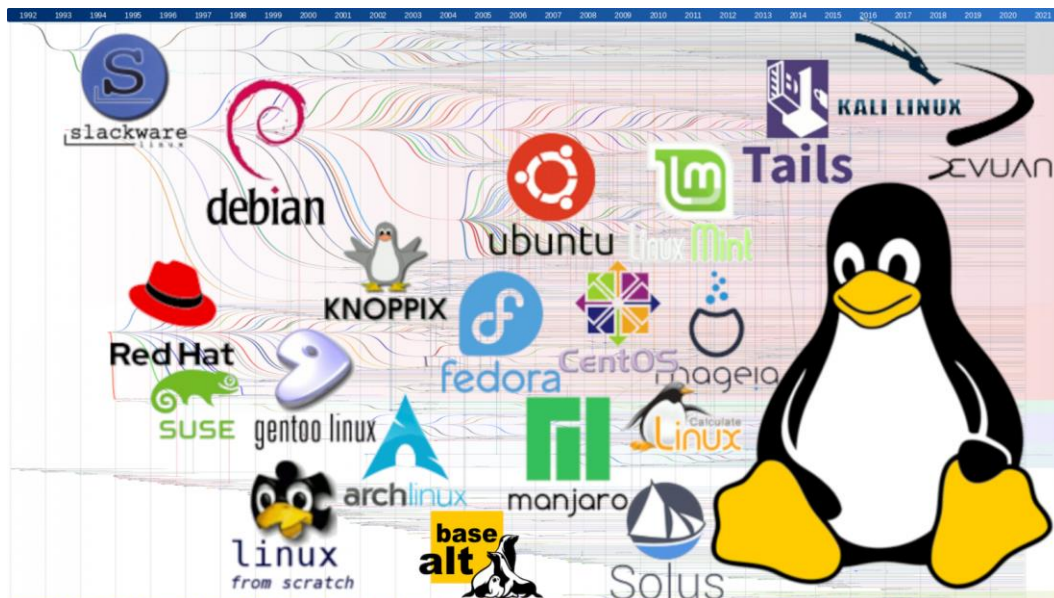


Рисунок 1.2 – Різновиди дистрибутивів Linux [5]

Всі ці фактори дозволяють мінімізувати навантаження на ресурси, що особливо важливо для серверів, які працюють цілодобово.

Для домашнього використання найчастіше застосовуються дистрибутиви Ubuntu Server, Debian, CentOS, Fedora Server, OpenSUSE або спрощені варіанти Xubuntu, Linux Mint тощо.

Ubuntu Server є одним із найпопулярніших варіантів завдяки стабільним оновленням, великій спільноті користувачів і зручній системі пакетів APT. Debian у свою чергу відзначається надзвичайною стабільністю та безпекою, тому часто використовується як основа для інших дистрибутивів. CentOS і Fedora Server є хорошим вибором для тих, хто хоче отримати функціональність корпоративного рівня з підтримкою сучасних технологій [6].

Переваги Linux полягають у високій безпеці, підтримці широкого спектра файлових систем, можливості роботи без графічного інтерфейсу, низькому енергоспоживанні та сумісності з більшістю серверного програмного забезпечення. Недоліком може бути складність початкового налаштування для недосвідчених користувачів, проте гнучкість і стабільність системи значно компенсують цей аспект.

1.3.2 Операційні системи Windows

Операційні системи сімейства Windows також широко використовуються для створення домашніх серверів особливо серед користувачів, які вже знайомі з екосистемою Microsoft. Найпоширенішими варіантами є Windows 10/11 Pro, Windows Server 2019/2022, а також Windows Home Server.

Перевагою Windows є інтуїтивно зрозумілий графічний інтерфейс, що спрощує налаштування системи навіть для користувачів без глибоких технічних знань. Вбудовані засоби, такі як «Диспетчер пристроїв», «Керування дисками», «Планувальник завдань» і служби Windows дозволяють швидко організувати спільний доступ до файлів, налаштувати віддалений робочий стіл, створити FTP або медіасервер за допомогою сторонніх програм [7].

Однак Windows має і певні недоліки. Один з найбільших недоліків – це високі системні вимоги. Окрім цього закритий код та потреба в регулярному ліцензуванні. Окрім цього робота системи з часом може сповільнюватися через накопичення оновлень і фонових процесів. Незважаючи на це, Windows залишається популярною серед користувачів, яким важлива зручність, сумісність із широким спектром прикладних програм і можливість інтеграції з іншими пристроями у домашній мережі.

1.3.3 Операційні системи для одноплатних комп'ютерів

Окрему категорію становлять легкі операційні системи призначені для енергоефективних і компактних пристроїв. Найвідомішим прикладом є Raspberry Pi OS, яка створена спеціально для одноплатних комп'ютерів Raspberry Pi. Ця ОС базується на Debian і оптимізована для ARM-архітектури, що забезпечує стабільну роботу навіть на обмежених апаратних ресурсах.

Raspberry Pi OS надає користувачам можливість легко розгорнути серверні служби для спільного доступу до файлів, обміну медіаконтентом або вебхостингу. Завдяки низькому енергоспоживанню пристрої Raspberry Pi можуть працювати безперервно. Користувач має змогу адмініструвати систему через SSH. Raspberry Pi OS підтримує більшість програм Linux, а завдяки своїй

простоті є відмінним варіантом для експериментів, навчання і створення прототипів [8].

При виборі операційної системи для домашнього сервера слід враховувати низку критеріїв, а також сумісність із необхідними програмними сервісами. Linux підходить для користувачів, які хочуть отримати максимальний контроль, надійність і гнучкість. Windows створена для тих, хто віддає перевагу зручності, сумісності з популярним ПЗ і простоті налаштування. Raspberry Pi OS дозволяє створити економічне, малогабаритне та безшумне рішення.

Варто зазначити, що з розвитком технологій межа між цими системами поступово стирається. Сучасні версії Linux мають зручні графічні оболонки, а Windows пропонує можливість запуску Linux-додатків через Windows Subsystem for Linux. Це відкриває нові можливості для поєднання різних екосистем у межах однієї домашньої інфраструктури.

1.4 Функціональні можливості домашніх серверів

Функціональні можливості домашніх серверів охоплюють широкий спектр завдань (рис. 1.3). Починаючи базовим зберігання даних і закінчуючи складними мережевими і мультимедійними сервісами, що забезпечують повноцінну взаємодію між усіма пристроями у межах домашньої мережі. Завдяки розвитку апаратних і програмних технологій домашні сервери поступово перетворилися з простих сховищ файлів на потужні універсальні системи, які здатні задовольнити потреби багатьох користувачів. На сьогоднішній день, завдяки сучасним технологіям та рішенням, домашні сервери здатні виконувати десятки різних функцій одночасно. Проте їхні можливості не обмежуються лише класичними сценаріями використання. Сучасний домашній сервер може виступати основою для реалізації комерційних проєктів.



Рисунок 1.3 – Застосування сервера

Основна перевага домашніх серверів полягає у можливості централізації інформації і сервісів, що робить управління цифровим простором користувача зручнішим, більш безпечним і контрольованим [9].

Найпоширенішою функцією домашнього сервера є зберігання даних. Завдяки цьому користувач може створити єдине сховище для документів, фото, відео, резервних копій, ігор чи інших файлів до якого мають доступ усі пристрої мережі. Замість дублювання одних і тих самих даних на різних комп'ютерах чи смартфонах усе зберігається на сервері, що спрощує організацію інформаційного простору. Доступ до файлів може здійснюватися через стандартні протоколи або через веб-інтерфейс. Такий підхід не лише підвищує зручність користування, а й забезпечує більшу безпеку оскільки всі файли залишаються в межах локальної мережі і не передаються стороннім хмарним сервісам. Окрім цього домашні сервери часто використовуються, як аналоги для хмарних рішень. Програми типу Nextcloud або ownCloud дозволяють створити власну хмару, яка не уступає по функціоналу Google Drive чи Dropbox, але з повним контролем над даними. Користувачі можуть завантажувати файли, ділитися ними через посилання, редагувати документи

онлайн, синхронізувати фото з мобільних пристроїв тощо. Така функціональність робить домашній сервер незалежним від зовнішніх постачальників послуг і гарантує приватність.

Другою ключовою функцією є резервне копіювання. Сучасні домашні сервери без проблем здатні автоматично створювати копії даних із комп'ютерів, телефонів або інших пристроїв, що підключені до мережі. Це дозволяє відновити інформацію у разі втрати, пошкодження чи збоїв [10]. Існують різні способи резервного копіювання повне, інкрементне або диференціальне.

Повне резервне копіювання створює копії усіх обраних даних за один цикл і поміщає в один файл. Це найпростіший і найнадійніший тип резервного копіювання, проте для реалізації такого копіювання потрібно витратити значну кількість часу і мати достатньо велике сховище. В процесі диференціального резервного копіювання копіюються усі файли, які були змінені або створені з моменту останнього повного резервного копіювання. Такий спосіб швидший, ніж повне копіювання, і займає менше місця. Проте обсяг копії збільшується з кожним днем, для відновлення потрібно мати повну резервну копію і останню диференціальну. Інкрементне резервне копіювання кардинально відрізняється від повного, але є схожим на диференціальне. Для реалізації інкрементного резервного копіювання слід спершу провести повне резервне копіювання [11]. Після цього будуть створюватись копії лише тих файлів, які були змінені з моменту останнього резервного копіювання. Цей спосіб найшвидший і займає найменше місця в сховищі зберігання, проте він найбільш складний в реалізації. Окрім цього для відновлення даних потрібно мати повне резервне копіювання та всі наступні інкременти копії. Для реалізації таких процесів використовуються спеціалізовані програмні засоби або спеціальні готові рішення NAS-пристрої. Резервне копіювання особливо актуальне в умовах зростаючої кількості кібератак, помилок користувачів і фізичних поломок накопичувачів.

Ще однією важливою функцією є потокова передача контенту. Користувач може створити власний мультимедійний центр, який забезпечує потокову трансляцію відео, музики чи фотографій на інші пристрої. Сервіси, які реалізують функцію медіастрімінгу, автоматично організують бібліотеки, додають метадані, обкладинки, описи фільмів і підтримують потокове відтворення через мережу. Це дозволяє створити персональну альтернативу комерційним платформам на зразок Netflix чи Spotify, де весь контент зберігається локально та контролюється власником сервера.

Домашній сервер може виконувати роль веб-сервера або бути в якості домашньої міні-лабораторії. За допомогою програм Apache, Nginx або Lighttpd можна розміщувати власні сайти, блоги чи локальні сервіси без необхідності оренди стороннього хостингу. Це особливо корисно для розробників та студентів, які бажають навчатися веб-адмініструванню або розгорнути внутрішні інформаційні портали. Веб-сервер у домашній мережі також може обслуговувати системи моніторингу або використовуватись для керування розумним будинком. Ще однією цікавою можливістю є контейнеризація та віртуалізація. Завдяки технологіям Docker або VirtualBox користувач може запускати кілька ізольованих середовищ, кожне з яких виконує окрему задачу [12]. Це дозволяє ефективно розподіляти ресурси, підвищує безпеку й гнучкість при тестуванні нових сервісів або конфігурацій. У сучасних умовах дедалі частіше домашні сервери інтегруються з системами розумного дому. На базі таких рішень, як Home Assistant, OpenHAB або Domoticz, сервер може збирати дані з датчиків, керувати освітленням, кліматом, безпекою та побутовими приладами. Завдяки інтеграції з голосовими асистентами користувач отримує централізоване управління всіма аспектами домашньої автоматизації через один інтерфейс.

Ще одна поширена функція – це створення і керування спільним доступом до принтерів та інших периферійних пристроїв. Сервер може виступати в ролі сервера для друку, що дозволяє будь-якому користувачу локальної мережі надсилати документи на друк без необхідності підключення

до принтера напряму. Це зручно для сімей або малих офісів де один пристрій використовується кількома людьми одночасно.

Серед розширених можливостей важливе місце займає керування мережею та підвищення безпеки. Домашній сервер може виконувати роль маршрутизатора, DNS, DHCP або проксі-сервера. Це дозволяє контролювати підключення пристроїв, фільтрувати трафік, створювати обмеження доступу, організовувати VPN-з'єднання для безпечної роботи з публічних мереж або забезпечувати віддалений доступ до домашньої інфраструктури. Завдяки VPN сервер може створювати зашифрований канал зв'язку між користувачем і домашньою мережею, що гарантує конфіденційність переданих даних.

Важливою функцією є моніторинг та аналітика системи. Сервер може збирати статистику про використання ресурсів, температуру процесора, мережеве навантаження, а також вести журнали подій. Інструменти типу Grafana, Zabbix або Netdata дозволяють у реальному часі спостерігати за станом системи, що допомагає своєчасно виявляти несправності або перевантаження. Такі можливості наближають домашні сервери за функціоналом до професійних серверних систем.

Ще однією перспективною функцією є підтримка систем відеоспостереження. Сучасний ринок пристроїв для відеоспостереження пропонує надзвичайно широкий вибір готових рішень. Дані рішення орієнтовані, як на приватних користувачів, так і на корпоративний сектор. Аналіз таких систем дозволяє оцінити рівень розвитку технологій, зручність впровадження, співвідношення ціни та функціональності. Також визначити оптимальний варіант для використання в домашній системі безпеки. Готові рішення сьогодні можуть бути реалізовані у вигляді окремих пристроїв наприклад, однієї IP-камери з хмарним зберіганням, або ж у вигляді комплектів відеоспостереження, що включають декілька камер, реєстратор і програмне забезпечення для моніторингу. Домашній сервер може виступати як система зберігання та обробки відео з камер. Така функціональність особливо важлива для забезпечення безпеки приватного будинку або квартири.

1.5 Безпека та моніторинг

Безпека та моніторинг є одним із найважливіших аспектів під час створення та експлуатації домашнього сервера. Вони є фундаментальними складовими будь якого домашнього серверного середовища, адже саме вони визначають стабільність, надійність і захист усієї цифрової інфраструктури. У сучасних умовах більшість пристроїв постійно підключені до мережі, а обсяг особистих даних безперервно зростає, саме тому питання інформаційної безпеки стає пріоритетним. Домашній сервер, що зберігає персональні файли, медіаконтент, резервні копії, елементи системи розумного дому і навіть відео з камер відеоспостереження потребує комплексного підходу до захисту. Основним завданням є не лише запобігання несанкціонованому доступу, але й постійний контроль стану системи, вчасне виявлення потенційних загроз і забезпечення безперервної роботи сервісів.

Загалом безпеку домашнього сервера можна розділити на кілька рівнів: фізичний захист, мережевий захист, системна безпека операційної системи, захист даних і моніторинг активності. Кожен із цих рівнів виконує свою роль, але всі вони взаємопов'язані і формують єдину систему захисту [13].

Першим кроком є захист самого пристрою від несанкціонованого фізичного доступу. Сервер повинен розміщуватися у безпечному місці, недоступному для сторонніх осіб. Важливо забезпечити належні умови охолодження, уникати перепадів напруги за допомогою джерел безперебійного живлення. Це не лише знижує ризик поломки, але й запобігає втраті даних у випадку раптового вимкнення живлення. У разі використання одноплатних систем необхідно подбати про захист SD-карти або SSD-накопичувача від пошкоджень. Не менш важливо забезпечити захист від інцидентів природнього походження.

Найвразливішим аспектом пристрою є його підключення до мережі. Саме через мережу здійснюються більшість атак, тому правильна конфігурація мережевого середовища – це основа кібербезпеки. Одним із найважливіших

інструментів захисту є брандмауер, який контролює вхідний та вихідний трафік. У Linux це зазвичай iptables або сучасніший інтерфейс UFW, а у Windows – вбудований Windows Defender та Firewall. За допомогою брандмауера можна обмежити доступ до серверу лише з локальної мережі або дозволити лише певні порти й протоколи.

Також важливо реалізувати сегментацію мережі, створюючи окремі підмережі для різних типів пристроїв. Це знижує ризик проникнення з менш захищеного сегмента до основної серверної частини. Використання VPN-з'єднання дозволяє безпечно підключатися до домашнього сервера ззовні, створюючи зашифрований канал зв'язку між користувачем і внутрішньою мережею. Програми типу OpenVPN або WireGuard забезпечують надійний рівень шифрування та автентифікації.

Операційна система повинна бути оновлена до актуальної версії з усіма останніми виправленнями безпеки. Багато користувачів нехтують цим аспектом, але саме застаріле програмне забезпечення найчастіше є точкою входу для зломисників. У Linux це реалізується через регулярні оновлення пакетів, а у Windows – через систему Windows Update.

Окрім цього необхідно обмежити кількість користувачів, які мають доступ до системи та створювати окремі облікові записи з різними правами доступу. Використання паролів високої складності значно зменшує ризик несанкціонованого входу. Для додаткового рівня захисту можна впровадити двохфакторну автентифікацію при віддаленому підключенні.

Ще один важливий аспект – це обмеження служб і процесів, які запускаються автоматично. Кожен активний сервіс потенційно є точкою входу для зломисників. саме тому варто вимикати або видаляти непотрібні служби і залишати лише ті, що дійсно необхідні. Також рекомендується використовувати інструменти контролю безпеки, наприклад Fail2Ban, який автоматично блокує IP-адреси після кількох невдалих спроб входу, запобігаючи brute-force атакам.

Одним із найефективніших способів забезпечити конфіденційність інформації є шифрування. Це може бути як шифрування всього диску, так і окремих каталогів або резервних копій. У разі крадіжки пристрою або накопичувача стороння особа не зможе отримати доступ до вмісту без ключа дешифрування.

Для забезпечення цілісності даних важливо використовувати системи резервного копіювання з перевіркою контрольних сум. Це дозволяє виявляти зміни або пошкодження файлів. Також доцільно мати декілька копій критично важливої інформації. Одну копію можна мати на локальному диску, іншу на зовнішньому накопичувачі чи у віддаленій хмарі з шифруванням.

Моніторинг – це постійне спостереження за роботою сервера, яке дозволяє вчасно виявити несправності, перевантаження або підозрілу активність. Це стосується як апаратних параметрів, так і мережевої активності. Для цього використовуються спеціальні системи моніторингу, такі як Zabbix, Netdata, Grafana, Prometheus, Nagios (рис. 1.4).



Рисунок 1.4 – Робочий інтерфейс Zabbix [14]

Вони збирають дані з різних джерел і відображають їх у вигляді графіків, таблиць чи дашбордів. Наприклад, адміністратор може бачити, коли сервер

починає споживати надмірно багато ресурсів або коли температура процесора перевищує допустимі межі. Додатково можна налаштувати систему сповіщень, яка буде сповіщати, коли будуть відбуватись певні сценарії. Також важливо вести журнали подій. У Linux цим займається служба `systemd-journald` або `rsyslog`, у Windows – системний журнал подій. Аналіз логів дозволяє простежити хто і коли підключався до системи, які процеси запускалися, чи були помилки або попередження.

Якщо сервер виконує роль веб-сервера, медіа-сервера або системи віддаленого доступу, необхідно забезпечити захист кожного з цих сервісів окремо. Для вебінтерфейсів слід використовувати HTTPS-з'єднання з надійними сертифікатами. Усі можливості адміністратора повинні бути захищені надійним паролем і доступні лише з внутрішньої мережі або через VPN.

1.6 Аналіз готових рішень

Сучасний ринок пропонує велику кількість готових рішень для створення та організації домашнього сервера, які відрізняються функціональністю, апаратними можливостями, рівнем складності налаштування та ціною. Завдяки цим рішенням користувач може не лише заощадити час на розгортанні інфраструктури, але і отримати стабільну та перевірену систему, що виконує конкретні завдання. Аналіз існуючих платформ дозволяє зрозуміти, які рішення сьогодні домінують у сфері домашніх серверів, які їхні переваги та обмеження, а також які технології найкраще підходять для подальшого вдосконалення чи модернізації власної серверної системи.

Одним із найпоширеніших рішень є NAS. NAS – це спеціалізовані пристрої для зберігання та обміну даними через мережу (рис. 1.5).



Рисунок 1.5 – NAS сервер [15]

Виробники пропонують різноманітні моделі NAS, які дозволяють організувати домашнє сховище файлів, створювати резервні копії, надавати віддалений доступ і навіть запускати мультимедійні сервери. Сучасні NAS пристрої оснащуються зручними вебінтерфейсами, мають підтримку RAID-масивів, що забезпечує надійність зберігання, а також підтримують інтеграцію з хмарними платформами. Synology DiskStation має власну операційну систему DSM, яка включає безліч вбудованих програм. Основною перевагою таких систем є простота налаштування та стабільність роботи, однак їхній головний недолік висока вартість і обмежена гнучкість [16].

Іншим підходом є використання одноплатних комп'ютерів, таких як Raspberry Pi, для побудови бюджетного і функціонального домашнього сервера. Raspberry Pi має низьке енергоспоживання, компактні розміри та підтримку численних дистрибутивів Linux. Завдяки цьому пристрій можна легко перетворити на сервер, який буде використовуватись в особистих цілях. Наприклад, дистрибутив Raspberry Pi OS підтримує повний набір серверних інструментів, включаючи SSH-доступ, Apache, MySQL, Docker, що робить його ідеальним рішенням для дому. Основна перевага такого підходу гнучкість і низька вартість, окрім цього такий пристрій малий в розмірах та невибагливий

у користуванні, однак продуктивність обмежена можливостями апаратного забезпечення.

OpenMediaVault – це готова система для побудови NAS-сервера на базі Debian. Вона має інтуїтивний вебінтерфейс, який дозволяє створювати спільні теки, налаштовувати RAID, резервне копіювання, DLNA і навіть torrent-завантаження без ручного редагування конфігурацій. OMV добре підходить для домашнього використання та не потребує високих знань з інформаційних технологій [17].

TrueNAS CORE рішення, що базується на FreeBSD. Його основною особливістю є файловий система ZFS, яка забезпечує високу надійність, контроль цілісності даних і розширені можливості резервного копіювання. TrueNAS орієнтований на користувачів, яким потрібна максимальна стабільність і захист даних, однак його налаштування потребує певного рівня технічних знань.

Для користувачів, які прагнуть поєднати функції сховища, мультимедійного центру та хмарного середовища, ідеальним вибором можуть стати системи Plex, Jellyfin або Emby. Plex є комерційним продуктом із розвиненою екосистемою, який дозволяє організувати власну бібліотеку медіафайлів з можливістю потокового відтворення на будь-якому пристрої. Jellyfin, в свою чергу, є повністю безкоштовною та відкритою альтернативою, яка підтримує ті самі функції, але не потребує підписки й не надсилає дані третім сторонам. Emby поєднує у собі обидва підходи, пропонуючи базову безкоштовну версію та додаткові платні можливості. Усі три рішення забезпечують автоматичну організацію контенту, додавання метаданих, субтитрів та доступ із будь-якої точки світу, що робить їх бажаними компонентами мультимедійного домашнього сервера.

Окрему увагу слід приділити Nextcloud і ownCloud. Ці платформи дозволяють розгорнути приватну хмару для зберігання та синхронізації даних. Вони надають функції обміну файлами, календарів, контактів, документів, мають підтримку розширень і мобільних додатків. Nextcloud пропонує

інтеграцію з офісними пакетами, що дозволяє редагувати документи онлайн, подібно до Google Docs. Таким чином, користувач отримує власний хмарний сервіс із повним контролем над інформацією без ризику витоку даних у публічні хмари [18].

Існують також комплексні готові рішення, що поєднують кілька функцій одночасно. Наприклад, CasaOS, YunoHost або Home Server OS – це програмні платформи, які надають графічний інтерфейс для керування додатками, контейнерами, медіасервісами, хмарними сховищами та вебсерверами. Їхня головна перевага полягає у простоті розгортання. Користувач може встановити систему одним кліком і швидко додати необхідні сервіси через інтегрований магазин застосунків. Такі рішення особливо зручні для початківців, які хочуть отримати функціональний домашній сервер без глибоких технічних знань.

У цілому, аналіз готових рішень демонструє, що сучасні технології дозволяють створити домашній сервер будь-якої складності в залежності від потреб та можливостей, від простого файлового сховища до повноцінної серверної інфраструктури з функціями віртуалізації, резервного копіювання та відеоспостереження. Ключовими критеріями вибору конкретного рішення є мета використання, наявне обладнання, рівень технічної підготовки користувача та бажаний ступінь автоматизації. Якщо потрібен простий і стабільний варіант варто звернути увагу на NAS-пристрої або OpenMediaVault. Якщо ж потрібна гнучкість, можливість масштабування та інтеграції з іншими сервісами, краще обрати Linux-дистрибутив із ручним налаштуванням або системи Raspberry Pi із власноруч конфігурованими компонентами.

Таким чином, готові рішення є основою для розуміння архітектури сучасних домашніх серверів. Вони дозволяють не лише швидко створити базову інфраструктуру, але й слугують відправною точкою для експериментів, оптимізації та розробки індивідуальних рішень, що можуть перевершувати за ефективністю й функціональністю комерційні аналоги. Саме на базі аналізу таких систем можливо сформулювати підхід до побудови модернізованої,

оптимізованої серверної інфраструктури, адаптованої під конкретні потреби користувача.

РОЗДІЛ 2

АНАЛІЗ ТА ПРОЕКТУВАННЯ ДОМАШНЬОЇ СЕРВЕРНОЇ ІНФРАСТРУКТУРИ

2.1 Моделі побудови серверної інфраструктури

Ще одним кроком при проектуванні домашньої серверної інфраструктури є визначення архітектурної моделі. Архітектурна модель буде визначати спосіб організації обчислювальних ресурсів, взаємодію компонентів, масштабованість та ефективність використання системи. Від правильно обраної моделі побудови залежить не лише продуктивність і стабільність роботи серверу, але й гнучкість у подальшому розвитку, оновленні та інтеграції нових сервісів. В сучасній практиці найчастіше застосовуються чотири базові підходи: монолітна, модульна, віртуалізована та кластерна моделі інфраструктури. Кожна з них має свої переваги та недоліки, обмеження та особливості розгортання, а також сфери застосування.

Монолітна модель є найпростішою та найстарішою формою побудови серверної інфраструктури. Вона передбачає, що всі служби додатки та сервіси розгортаються на одному фізичному пристрої, який одночасно виконує всі функції системи. Всі компоненти мають спільну операційну систему, спільну файлову структуру та апаратні ресурси. Основною перевагою монолітного підходу є простота реалізації та налаштування. Для розгортання достатньо встановити одну операційну систему та налаштувати відповідні служби. Усе функціонує в межах одного середовища, що знижує складність адміністрування. Такий підхід ідеально підходить для початкових етапів створення домашнього сервера або для користувачів, які не мають значного досвіду в системному адмініструванні. Ще однією перевагою монолітної архітектури є мінімальні вимоги до апаратного забезпечення. Для неї не потрібно мати потужний багатоядерний процесор або великі обсяги оперативної пам'яті. Це робить монолітну модель оптимальним варіантом для бюджетних домашніх рішень або використання старих комп'ютерів. До

недоліків цієї моделі можна віднести проблеми при розширенні функціональності. Низька гнучкість – головна проблема монолітної моделі. Усі служби тісно пов'язані між собою, тому оновлення або зміна одного компонента може призвести до конфліктів або збоїв у роботі інших. Також у випадку збою всієї системи або пошкодження ОС перестають працювати всі сервіси одночасно. Окрім цього масштабування в монолітній моделі є обмеженим. Неможливо додати потужності або розділити навантаження між різними пристроями. Коли навантаження сильно зростає, тоді система стає нестабільною. Монолітна модель доцільна у випадках, коли домашній сервер виконує одну або декілька нескладних функцій для особистих потреб. Вона характеризується стабільністю, передбачуваністю та простотою, але має обмеження у розвитку та масштабуванні.

Модульна модель базується на принципі поділу функцій між окремими компонентами або пристроями, кожен із яких виконує певну роль у загальній системі. Наприклад, один пристрій може відповідати за зберігання даних, інший за мультимедійні функції, а третій – за резервне копіювання або моніторинг. Такий підхід забезпечує вищу гнучкість, надійність і масштабованість. Кожен модуль може бути розгорнутий окремо, оновлюватися незалежно від інших, а в разі збою одного компонента інші продовжують працювати. Це дозволяє проводити технічне обслуговування або модернізацію без повної зупинки системи. Модульна архітектура є логічним етапом еволюції після монолітної. У ній застосовується принцип «розділення відповідальності» – кожен компонент виконує лише свої завдання, а комунікація між ними здійснюється через мережу. Наприклад, файловий сервер може бути побудований на базі OpenMediaVault, медіасервер – на Plex, контролер розумного будинку – на Home Assistant, при цьому всі вони об'єднані у спільну локальну мережу. До переваг можна віднести простоту масштабування. Можна додати новий пристрій або компонент, який виконуватиме окрему функцію без необхідності змінювати всю систему. Вихід з ладу одного вузла не призводить до зупинки всієї інфраструктури, що

забезпечує високу відмовостійкість. Завдяки гнучкості до модернізації можна оновлювати або замінювати окремі модулі не впливаючи на інші. Однак існують і певні недоліки. По-перше, модульна інфраструктура потребує глибшого розуміння мережевої взаємодії та правильного налаштування комунікацій між компонентами. По-друге, така система зазвичай вимагає більшої кількості пристроїв, що підвищує енергоспоживання та складність управління. У домашніх умовах модульний підхід дозволяє реалізувати збалансовану інфраструктуру у якій кожен пристрій виконує свою чітко визначену роль. Потужний комп'ютер може виступати головним сервером, Raspberry Pi як медіацентр, а старий ноутбук – як резервна станція. Така гнучкість робить модульну модель одним із найпоширеніших підходів у сучасних домашніх серверних системах.

Кластерна модель є більш складною і зазвичай використовується у корпоративних або високонавантажених середовищах. Її принципи роботи можна адаптувати і для домашнього використання. Суть підходу полягає в об'єднанні кількох фізичних або віртуальних серверів в єдину систему, яка функціонує як один потужний пристрій. У кластері обчислювальні ресурси розподіляються між вузлами, що дозволяє підвищити продуктивність, забезпечити балансування навантаження та відмовостійкість. Якщо один вузол виходить із ладу, його навантаження автоматично перерозподіляються між іншими. Існує кілька типів кластерів:

- обчислювальні, які орієнтовані на паралельні обчислення;
- серверні кластери, що забезпечують високу доступність сервісів;
- файлові кластери, що створюють єдине спільне сховище з доступом із різних вузлів.

У домашній інфраструктурі кластерна модель може застосовуватися для організації резервного дублювання, балансування між кількома серверами або навіть для створення власної міні-хмари. Основними перевагами кластера є:

- висока відмовостійкість – при відмові одного вузла система продовжує працювати;

- масштабованість, яка дозволяє легко додавати нові вузли для збільшення потужності;

- підвищена продуктивність за рахунок паралельної роботи кількох серверів.

Головний недолік полягає в складності реалізації. Налаштування кластерів потребує знань мережевої інфраструктури, розподілених файлових систем і балансувальників навантаження. Окрім цього для ефективної роботи потрібна стабільна і швидка локальна мережа. Проте навіть спрощені варіанти кластера можуть знайти місце в домашніх умовах. Наприклад, Proxmox Cluster дозволяє об'єднати кілька вузлів із різних пристроїв у єдине кероване середовище, що полегшує управління і підвищує надійність.

Віртуалізована модель – це сучасний підхід, який полягає в розгортанні кількох ізольованих віртуальних середовищ, а саме віртуальних машин або контейнерів на одному фізичному пристрої. Кожне таке середовище має власну операційну систему, ресурси та виконує певну функцію. Віртуалізація дозволяє максимально ефективно використовувати ресурси одного пристрою. Замість трьох фізичних комп'ютерів можна запустити три віртуальні сервери на одному потужному пристрої. Для реалізації цього підходу використовують гіпервізори типу VirtualBox, VMware, Proxmox VE, Hyper-V, або системи контейнеризації Docker, LXC, Podman. Основна перевага віртуалізованої архітектури полягає у гнучкості та ізоляції. Кожен віртуальний сервер функціонує незалежно, що дозволяє легко тестувати нові сервіси, експериментувати з конфігураціями, створювати резервні копії у вигляді snapshot. Окрім цього, у разі збоїв можна швидко відновити роботу лише конкретної віртуальної машини. Віртуалізація також відкриває можливість оптимізації використання ресурсів. Неактивні віртуальні машини можуть бути призупинені, а ресурси перерозподілені на активні сервіси. Це особливо корисно у домашньому середовищі де навантаження змінюється залежно від часу доби чи активності користувачів. Проте віртуалізована модель має певні недоліки. По-перше, вона потребує потужного апаратного забезпечення. Багатоядерний процесор, великий обсягу

оперативної пам'яті та швидкий накопичувач. По-друге, сама віртуалізація використовує певну кількість апаратних ресурсів для підтримання своєї працездатності, тому ефективність може бути нижчою ніж при безпосередньому запуску сервісів на модульній системі. Віртуалізована модель вважається оптимальним вибором для домашніх серверів, які виконують багато різноманітних функцій одночасно. Вона поєднує гнучкість, безпеку та масштабованість, що робить її популярною серед розробників.

2.2 Критерії вибору операційної системи

Вибір операційної системи є одним із найважливіших етапів під час створення та налаштування домашнього сервера оскільки саме ОС визначає стабільність, продуктивність, функціональні можливості, рівень безпеки та зручність адміністрування системи. Правильне рішення у цьому питанні дозволяє не тільки оптимізувати роботу апаратної частини, а ще й забезпечити гнучке керування всіма сервісами, що розміщуються на сервері.

Оскільки при виборі операційної системи всі ці критерії впливають на ефективність роботи з системою та на саму ефективність системи, для домашнього сервера необхідно враховувати:

- продуктивність та використання ресурсів;
- сумісність з апаратним забезпеченням;
- стабільність та надійність;
- безпека та оновлення;
- простота адміністрування;
- підтримка програмних сервісів;
- масштабованість і розширюваність;
- спільнота та технічна підтримка.

Домашні сервери часто працюють на обладнанні з обмеженими ресурсами. Часто це одноплатні комп'ютери або старі ноутбуки, які переобладнуються для серверних потреб. Тому важливо, щоб операційна

система мала оптимізоване ядро, споживала мінімум оперативної пам'яті та процесорного часу. Легкі дистрибутиви Linux мають суттєву перевагу в цьому аспекті над системами Windows, які вимагають більших ресурсів для стабільної роботи.

Не всі операційні системи підтримують широкий спектр обладнання. Для Raspberry Pi або інших ARM-пристроїв потрібні спеціалізовані апаратні конфігурації. Тоді як Windows орієнтований переважно на x86-архітектуру. Важливо також перевіряти підтримку драйверів, особливо якщо сервер планується розгорнути на старішому обладнанні.

Для серверних рішень надзвичайно важлива стабільність. Операційна система повинна забезпечувати безперебійну роботу протягом тривалого часу без необхідності частих перезавантажень або оновлень, що можуть викликати збої. Дистрибутиви Linux відомі своєю надійністю й стабільністю при тривалій роботі, тоді як Windows також пропонує стабільну роботу, але частіше потребує оновлень і перезавантажень.

Оскільки домашній сервер часто має доступ до мережі Інтернет, критично важливо, щоб операційна система мала розвинену систему безпеки. Контроль прав доступу, фаєрвол, шифрування, журналювання подій – це елементи, без яких система не матиме повноцінної безпеки. Системи Linux надають розширені інструменти безпеки тоді як Windows забезпечує безпеку через централізовану політику доступу, антивірусну інтеграцію та автоматичні оновлення.

Рівень підготовки користувача має велике значення. Якщо адміністратор не має досвіду роботи з командним рядком, то більш зручним вибором може стати Windows або Ubuntu із графічним інтерфейсом. Натомість дистрибутиви Linux пропонують більшу гнучкість і контроль, але вимагають знання системних команд і структури конфігураційних файлів.

Операційна система повинна забезпечувати сумісність із потрібними серверними застосунками та протоколами. Linux має перевагу у цьому аспекті, оскільки більшість серверного програмного забезпечення спочатку створюється

саме для Unix-подібних систем. Windows пропонує власні аналоги, які нічим не поступаються програмним забезпеченням, створеним для Unix-подібних систем.

Залежно від майбутнього розширення інфраструктури, слід обирати ОС, яка дозволяє легко додавати нові компоненти, сервіси або віртуальні машини. Linux завдяки відкритому коду і широкій підтримці контейнеризації краще підходить для масштабування.

Наявність активної спільноти користувачів, документації та форумів спрощує вирішення технічних проблем. Дистрибутиви Linux мають величезну кількість користувачів і доступних навчальних матеріалів, тоді як Windows орієнтований на офіційну підтримку Microsoft.

Для порівняння взяті операційні системи, що найчастіше використовуються для побудови домашніх серверів: Windows, Linux (Xubuntu) та Raspberry Pi OS.

Файлова система є одним із ключових елементів, що визначає стабільність, швидкість доступу до даних і можливість їх відновлення у разі збоїв. Windows використовує файлові системи NTFS і ReFS. NTFS підтримує розширені атрибути, шифрування, контроль доступу і журналювання. ReFS призначена для підвищеної стійкості до помилок, але рідше застосовується в домашніх умовах. Linux працює з файловими системами EXT4, Btrfs, XFS, які мають високу надійність і стійкість до збоїв. Окрім цього Btrfs підтримує функцію snapshots і динамічне розширення томів, що є перевагою для резервного копіювання. Raspberry Pi OS в міру своїх апаратних можливостей переважно використовує EXT4, яка добре оптимізована для флеш-пам'яті та карт microSD [19].

Windows добре масштабується у корпоративних середовищах, однак для домашнього використання такий рівень масштабування не є затребуваним і часто ніяк не реалізується. Дистрибутиви Linux забезпечують відмінну масштабованість як у напрямку ресурсів, так і за кількістю користувачів і сервісів. Raspberry Pi OS має обмежену масштабованість через апаратні

обмеження, але чудово підходить для невеликих серверних проєктів і тестових середовищ.

Windows Server пропонує вбудовану підтримку Active Directory, SMB, RDP, FTP, IIS, що зручно для інтеграції у гібридні мережі. Linux підтримує широкий спектр мережеслужб: SSH, FTP, NFS, SMB, Apache, Nginx, DLNA, VPN, Docker [20]. Його перевагою є відкритість і можливість розширення. Raspberry Pi OS підтримує практично всі сервіси Linux, але з урахуванням обмежених ресурсів пристрою.

Linux системи з моменту створення вважаються більш захищеними, оскільки кожна дія виконується лише від імені користувача з певними правами, а дії від імені адміністратора вимагають підтвердження. Окрім цього в Linux існує велика кількість програмного забезпечення для моніторингу безпеки. Windows має свої інструменти: BitLocker, Windows Defender, політики безпеки груп, але вони часто вимагають ручного налаштування для досягнення максимальної ефективності.

Windows має інтуїтивно зрозумілий графічний інтерфейс, що робить її зручною для користувачів без досвіду системного адміністрування. Linux та Raspberry Pi OS, хоча і мають графічні оболонки, проте найчастіше налаштовується через командний рядок, що потребує базових знань системних команд.

Linux і Raspberry Pi OS є повністю безкоштовними, що робить їх найпопулярнішими рішеннями для домашніх серверів. Windows є комерційним продуктом, який потребує ліцензії, але для навчальних або некомерційних цілей можна використовувати пробні версії із розширеними мережевими функціями.

2.3 Порівняння можливостей ОС

2.3.1 Особливості ОС

Для ефективно побудови домашньої серверної інфраструктури обрано та проведено аналіз трьох операційних систем Windows 10, Xubuntu, Raspberry Pi

OS. Кожна з цих систем має власну архітектуру, модель управління, файлові структури та особливості використання.

Операційна система є основою будь-якого серверного рішення, оскільки саме вона визначає, як будуть розподілятися апаратні ресурси між процесами. У серверах важливу роль відіграють такі параметри як стабільність, гнучкість, відмовостійкість, масштабованість та можливість безперервної роботи.

Windows 10 – це одна з найпопулярніших операційних систем. Розроблена компанією Microsoft, орієнтована на настільні ПК. Її сильні сторони – це високий рівень сумісності з програмами, дружній графічний інтерфейс та широкий спектр драйверів.

Xubuntu – це спрощений, але потужний дистрибутив Linux на базі Ubuntu, який оптимізований для пристроїв з середніми або слабкими апаратними ресурсами. Вона поєднує в собі стабільність ядра Linux з легкістю робочого середовища XFCE, що дозволяє ефективно використовувати апаратні ресурси.

Raspberry Pi OS – це спеціалізована система на базі Debian розроблена для ARM-процесорів. Вона відрізняється високою енергоефективністю та стабільністю при мінімальному споживанні ресурсів. Саме завдяки цим властивостям вона часто використовується для створення домашніх серверів.

2.3.2 Файлові системи

Ключовим компонентом будь-якої операційної системи є файлова система з якою вона працює. Саме від файлової системи залежить швидкість зчитування та записування даних, надійність збереження інформації, можливість відновлення після збою, а також підтримка прав доступу та журналювання.

Основною файловою системою для Windows 10 є NTFS. Цей тип файлових систем підтримує багато функцій, таких як: контроль доступу, шифрування і т.д. Окрім цього NTFS є досить надійною і це все завдяки журналюванню. Всі файли зберігаються в головній файловій таблиці, як файловий дескриптор, та як вміст із даними. Головна файлова таблиця записує всю інформацію про них. Бітова карта (BitMap) веде облік усіх вільних і використаних кластерів, журнал (Log), що застосовується при реєстрації

записів і файл BadClus, що містить інформацію про пошкоджені кластери, під це все резервується 16 перших записів в головній файлової таблиці. В даній файлової системі перший та останній сектори містять налаштування файлової системи.

Xubuntu працює на базі файлової системи EXT4. Вона поєднує в собі високу продуктивність, надійність і гнучкість. Окрім цього, EXT4 підтримує журналювання, великі об'єми даних, дефрагментацію на «живу» та відновлення після збою без втрати цілісності даних. Крім EXT4, Xubuntu може працювати з Btrfs, XFS та ZFS, що дозволяє оптимізувати файлову систему під конкретні серверні задачі.

Raspberry Pi OS використовує файлову систему EXT4 як основну для збереження даних, а завантажувальний розділ має формат FAT32, який забезпечує сумісність із BIOS-подібними середовищами. Такий підхід дозволяє системі швидко запускатися навіть на енергоефективних пристроях.

2.3.3 Масштабованість

Масштабованість – це здатність системи ефективно працювати зі зростанням навантаження, кількості користувачів або обсягу даних. Windows 10 чудово підходить для вирішення локальних завдань. При збільшенні кількості одночасних клієнтів або сервісів система швидко споживає більше ресурсів. Завдяки модульній архітектурі Linux ядро може динамічно керувати процесами. Завдяки чому Xubuntu має високий рівень масштабованості та гнучкості. Використання системного менеджера systemd дозволяє ефективно запускати, моніторити та перезапускати служби без втручання користувача. Також система підтримує контейнеризацію, що робить її придатною для різноманітних серверних проєктів. Raspberry Pi OS має обмежену масштабованість через апаратні обмеження самої плати, проте у межах своїх можливостей система працює дуже ефективно. Вона підтримує багатопотоковість, динамічне керування енергоспоживанням і легкі серверні служби.

Усі три системи підтримують основні серверні протоколи: SMB/CIFS, SSH, HTTP/HTTPS, FTP, DLNA, VPN. Windows 10 орієнтована на користувачів,

яким зручно працювати через графічний інтерфейс. Xubuntu і Raspberry Pi OS, навпаки, забезпечують повний контроль через термінал і відкриту архітектуру.

2.4 Критерії та вибір програмного забезпечення

Вибір програмного забезпечення для домашнього сервера є одним із ключових етапів його побудови, оскільки саме від встановлених сервісів і застосунків залежить функціональність, зручність користування, стабільність обміну даними та сумісність із різними пристроями. У сучасних умовах існує широкий спектр програм, які дозволяють організувати файловий доступ, потокову передачу медіа, резервне копіювання, моніторинг або автоматизацію.

Під час вибору програмного забезпечення для домашнього сервера слід враховувати ряд критеріїв. Програма має виконувати конкретне завдання. Повинна бути хороша сумісність із операційною системою. Для домашнього сервера бажано, щоб програмне забезпечення мало зрозумілий інтерфейс або можливість легко конфігуруватися. Чим менше навантаження створює сервіс, тим стабільніше працює система. Оскільки домашній сервер обслуговує пристрої різних типів, важливо щоб програми підтримували універсальні протоколи передачі даних.

Samba – це одне з найпоширеніших рішень для організації спільного доступу до файлів і принтерів у локальній мережі. Вона працює на основі протоколів SMB та CIFS, які спочатку розробила компанія Microsoft для Windows, але згодом їх підтримку було розширено і для UNIX-подібних систем.

Основне призначення Samba – забезпечити взаємодію між системами в спільній мережі. Завдяки цьому користувачі можуть отримувати доступ до файлів на сервері з будь-якого пристрою, який підтримує протокол SMB. Протокол SMB працює за моделлю клієнт-сервер, де клієнт надсилає запит до сервера для отримання доступу до файлів, перегляду каталогів, друку чи виконання інших дій. Samba використовує порти 137-139 і 445. Для

автентифікації вона використовує облікові записи користувачів. CIFS, як розширення SMB, підтримує додаткові функції, а саме роботу з Unicode, довгі імена файлів, контроль доступу, а також віддалене виконання команд. Завдяки відкритому коду Samba активно використовується на NAS-пристроях, Raspberry Pi і Linux серверах, забезпечуючи стабільну взаємодію з системами Windows без потреби у дорогому ліцензованому програмному забезпеченні [21].

До аналогів можна віднести NFS – протокол, розроблений для Unix-систем, що дозволяє монтувати віддалені каталоги як локальні. Працює швидше за SMB у Linux-середовищі, але має гіршу сумісність із Windows. FTP/SFTP – класичні протоколи для обміну файлами через клієнтські програми, однак не забезпечують безшовної інтеграції з файловою системою. WebDAV – розширення протоколу HTTP, що дозволяє працювати з файлами через веб-браузер або файловий менеджер.

MiniDLNA, також відомий як ReadyMedia, – це легкий медіасервер із відкритим кодом, який дозволяє передавати відео, аудіо та зображення на інші пристрої у домашній мережі. Він підтримує стандарти DLNA і UPnP. UPnP – це набір мережевих протоколів, який дозволяє пристроям автоматично виявляти одне одного в мережі та встановлювати між собою зв'язок без необхідності ручного налаштування. UPnP базується на стандартних інтернет-протоколах:

- SSDP використовується для виявлення пристроїв через UDP;
- HTTP для передачі описів пристроїв і сервісів;
- SOAP для керування пристроями та запиту інформації;
- DLNA є розширенням UPnP AV і визначає стандарти сумісності між пристроями.

У контексті домашнього сервера MiniDLNA виконує роль медіасервера, який сканує задані каталоги та дозволяє транслювати вміст для DLNA-клієнтів. Після запуску MiniDLNA автоматично оголошує свою присутність у локальній мережі за допомогою SSDP. Клієнт знаходить сервер, зчитує список доступних медіафайлів і відтворює їх через потокову передачу HTTP. До переваг MiniDLNA можна віднести мінімальне споживання ресурсів, підтримка великої

кількість пристроїв, відсутність складних налаштувань та автоматичне оновлення медіатеки при додаванні нових файлів [22].

Аналогами MiniDLNA є Plex Media Server – комерційний продукт, який підтримує транскодування відео, користувацькі профілі, віддалений доступ і мобільні додатки. Emby – схожий за функціоналом до Plex, з відкритою архітектурою та можливістю встановлення на Linux. Kodi – медіацентр із розширеними можливостями локального й мережевого відтворення контенту, часто використовується разом із DLNA.

Serviio – це медіасервер, який підтримує протоколи DLNA та UPnP. Його головна перевага полягає в тому, що він може транскодувати медіаконтент у реальному часі. Відповідно змінювати формат файлів, щоб пристрій, який не підтримує певний формат, міг без проблем відтворити відео або аудіо. Serviio підтримує різні типи пристроїв від Smart TV до ігрових консолей, мобільних телефонів та комп'ютерів.

Особливості роботи Serviio:

- сервер постійно сканує вказані каталоги й оновлює медіатеку;
- підтримує потокову передачу через HTTP;
- використовує UPnP/DLNA для виявлення клієнтів;
- підтримує метадані.

Serviio може керуватись через вебінтерфейс або додаток Serviio Console. Serviio використовує протоколи UPnP для автоматичного виявлення пристроїв. DLNA AV Transport – для передачі поточкових даних. HTTP – як основний транспортний механізм потокового передавання.

До аналогів Serviio можна віднести universal Media Server – відкритий аналог, побудований на базі старіших версій Serviio, підтримує Java. Plex – більш інтегроване рішення з акцентом на користувацький досвід і мобільний доступ. Jellyfin повністю безкоштовна альтернатива Plex із відкритим кодом, що активно розвивається у спільноті користувачів Linux.

РОЗДІЛ 3

ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ДОМАШНЬОЇ СЕРВЕРНОЇ ІНФРАСТРУКТУТИ

3.1 Планування дослідження

Для дослідження ефективності домашньої серверної інфраструктури заплановано серію експериментів. У процесі реалізації досліджень здійснено налаштування системи з фіксацією ключових параметрів:

- використання ресурсів (ЦП, ОЗУ) системою;
- температура компонентів;
- час ввімкнення та завантаження системи;
- енергоспоживання.

Дані параметри обрані для фіксації у зв'язку з тим, що саме вони є критичними для домашньої серверної інфраструктури. Використання ресурсів дозволяє оцінити, наскільки ефективно використовується обладнання, а надійність та відгук системи мають безпосередній вплив на користувацький досвід.

Дослідження проводились на комп'ютерному обладнанні, яке не використовувалось як серверна інфраструктура. Система побудована на основі фізичного пристрою, а саме ноутбук HP 250 G7, який використовувався як особистий комп'ютер. Серверні функції виконувалися на загальному рівні, без ізоляції служб, без централізованого моніторингу, резервного копіювання або захищеного віддаленого доступу.

Наявне обладнання:

- процесор: Intel Core i3-7020U (2 ядра, 2,3 ГГц);
- оперативна пам'ять: 8 ГБ DDR4;
- накопичувач: HDD 1 ТБ;
- графічний процесор: MDX110;
- графічний процесор: HD Graphics 620.

Окрім цього, задіяний мікрокомп'ютер Raspberry Pi 5 з процесором Cortex-A76, 8 Гб оперативної пам'яті та 64 Гб вбудованої.

Загалом застосовувалось три конфігурації. В першій конфігурації був застосований одноплатний комп'ютер Raspberry PI 5 з операційною системою Raspberry PI. Дана конфігурація вважається готовим рішенням оскільки операційна система було спеціально розроблена для лінійки одноплатних комп'ютерів, вона оптимізована для апаратного забезпечення Pi та має низькі вимоги до ресурсів. Другою конфігурацією є Windows 10 та ноутбук HP 250 G7, дана конфігурація має певну кількість аспектів які потребують оптимізації. З врахуванням того, що удосконалень потребує і апаратна і програмна частина, можна суттєво підвищити продуктивність та енергоефективність даної конфігурації. Третьою конфігурацією є Xubuntu ноутбук HP 250 G7. Xubuntu представляє собою вже готове рішення операційної системи проте дозволяє здійснити певні корективи для оптимізації.

Відповідно до цього в першій конфігурації виконається невелика оптимізація лише операційної системи. В другій конфігурації здійснюватимуться оптимізаційні і модернізаційні заходи над операційною системою та апаратним забезпеченням. В третій конфігурації виконається модернізація апаратної частини та невелика оптимізація ОС.

У ході досліджень моделювались такі сценарії використання, як: потокове відтворення мультимедіа, копіювання файлів через мережу, а також функціонування ігрового сервера. Це дозволить перевірити стабільність системи в умовах різного навантаження.

Для кращої ефективності дослідження дані записуватимуться та порівнюватимуться між собою. Для заміру навантаження на систему та температури компонентів використовувались інструменти моніторингу продуктивності такі як Диспетчер задач, AIDA64, htop. Енергоспоживання вимірювалось шляхом підключення пристрою до EcoFlow з можливістю відображення поточного енергоспоживання.

Дослідження всіх трьох конфігурацій відбувалось в однакових умовах, а саме: пристрої підключені до джерела живлення. Задіювалось по 5 пристроїв, які виступали в ролі користувачів і які користувались послугами сервера. Всі дії відбувались в локальній мережі. Окрім цього тестування відбувалось в трьох станах системи: стан спокою, стан середнього навантаження та стан максимально навантаження. Для точності отриманих даних здійснювалось по кілька тестувань. Під тестуванням мається на увазі копіювання даних на сервер або з сервера, використання ігрового сервера та перегляд медіаданих.

3.2 Підготовка пристрою

Підготовка пристрою є одним із ключових етапів перед початком розгортання домашньої серверної інфраструктури, оскільки саме від технічного стану апаратного забезпечення залежить стабільність, енергоефективність і надійність майбутньої системи. Для реалізації проєкту обрано ноутбук HP 250 G7, який виступає базовою платформою для побудови домашнього сервера. Перед безпосереднім розгортанням серверного середовища проведено комплекс робіт із технічного обслуговування та перевірки працездатності пристрою. Основне завдання на цьому етапі – забезпечити стабільне охолодження, достатній обсяг оперативної і постійної пам'яті та перевірити стан апаратних компонентів, які можуть вплинути на роботу сервера при тривалому навантаженні.

Першим кроком стала перевірка працездатності всіх основних компонентів системи, зокрема процесора, оперативної пам'яті, накопичувача та системи охолодження. За допомогою діагностичних інструментів перевірено працездатність процесора Intel Core i3-7020U, який має два фізичні ядра з тактовою частотою 2,3 ГГц. Результати тестування підтвердили його стабільну роботу без перегріву та збоїв під навантаженням. Також перевірено стан накопичувача HDD 1 ТБ, який продемонстрував відсутність пошкоджених секторів і допустимий рівень зношення.

Також проведено перевірку роботи акумулятора, оскільки навіть у стаціонарному режимі важливо мати резерв живлення на випадок короткочасного відключення електроенергії. Стан батареї виявився незадовільним і потребував заміни, окрім цього виявлено здуття батареї. Для уникнення займання або вибуху дану батарею замінено на нову.

Наступним етапом стала очистка системи охолодження, оскільки ноутбук протягом певного часу експлуатувався. Для стабільної роботи, де навантаження на процесор і жорсткий диск може бути постійним, здійснено повне очищення системи охолодження. Після очищення проведено заміну термопасти на центральному процесорі, оскільки старий термоінтерфейс втратив свої теплопровідні властивості. Дані дії необхідні, щоб уникнути перегріву в умовах безперервної роботи. Перегрівання процесора може призвести до зниження продуктивності або навіть до аварійного вимкнення системи.

Для підвищення ефективності роботи сервера також збільшено обсяг оперативної пам'яті(рис. 3.1).



Рисунок 3.1 – Ноутбук на початку процесу обслуговування та модернізації

Ноутбук мав 8 ГБ DDR4 оперативної пам'яті, однак з урахуванням планованого розгортання декількох служб одночасно, вирішено додати ще 4 ГБ оперативної пам'яті, щоб довести загальний обсяг до 12 ГБ DDR4 оперативної

пам'яті. Це дозволило покращити стабільність роботи та забезпечити комфортну багатозадачність.

З метою підвищення продуктивності системи встановлено додатковий накопичувач, а саме SSD з ємністю в 256 ГБ. Основною метою такого рішення є розділення функцій між різними типами накопичувачів SSD та HDD з урахуванням їх технічних особливостей. SSD використовується в якості системного диска, на якому розміщується операційна система та основне програмне забезпечення. Завдяки швидкості читання та запису даних, а також низькому часу доступу, підвищиться швидкодія системи і скоротиться час завантаження системи. В той час як HDD виконує функції основного сховища даних. Такий підхід дозволить отримати оптимальне поєднання швидкодії системи та великого обсягу пам'яті. Окрім цього розподіл ролей між накопичувачами підвищує загальну ефективність роботи пристрою, покращує стабільність системи та забезпечує швидкий доступ до найбільш затребуваних файлів та процесів.

Додатково налаштовано BIOS для оптимізації енергоспоживання (рис. 3.2), а саме, вимкнено функції, які не використовуватимуться. Додатково встановлено автоматичний запуск після відновлення живлення.

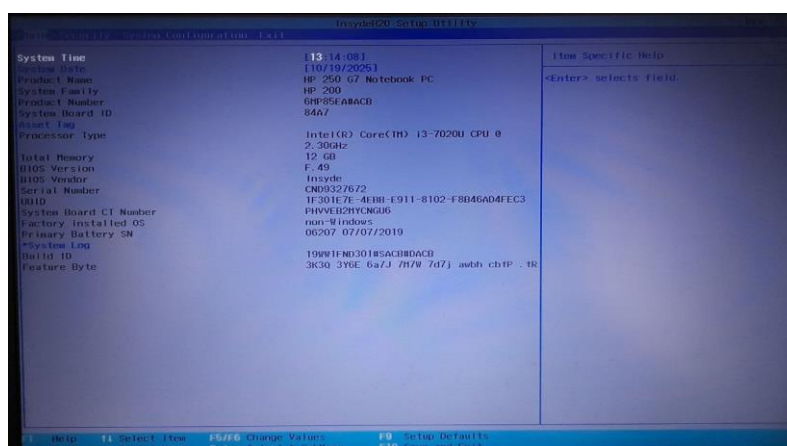


Рисунок 3.2 – Налаштування BIOS

Після всіх етапів пристрій HP 250 G7 продемонстрував стабільну роботу, низький рівень шуму й ефективне енергоспоживання. Це робить його

придатним для виконання функцій домашнього сервера і здатним працювати безперервно протягом тривалого часу.

3.3 Первинне налаштування конфігурацій

Етап первинного налаштування конфігурації є не менш важливим у процесі розгортання домашнього сервера, ніж інші етапи, оскільки саме на цьому етапі відбувається встановлення операційної системи, налаштування мережевих параметрів і базових служб. У рамках даної роботи проводилось встановлення та налаштування кількох операційних систем: Windows 10, Xubuntu на пристрій ноутбук HP 250 G7 та Raspberry Pi OS на пристрій Raspberry Pi 5, що дозволило порівняти їхні можливості, стабільність і зручність у контексті використання в домашній серверній інфраструктурі.

Для встановлення Windows 10 підготовлено завантажувальний USB-накопичувач за допомогою офіційного інструмента Microsoft Media Creation Tool (рис. 3.3).

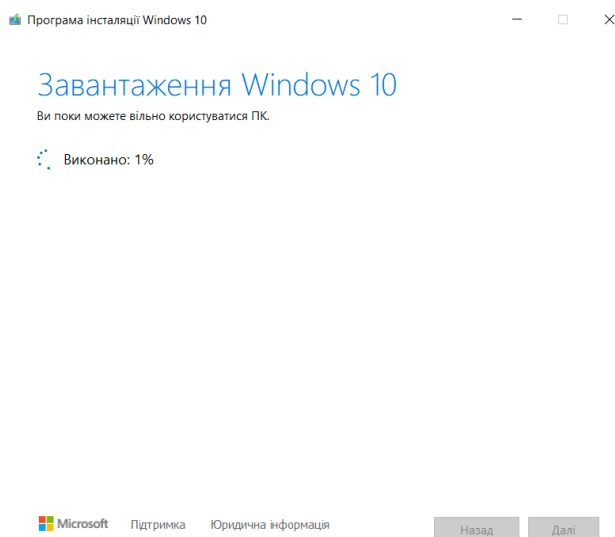


Рисунок 3.3 – Створення завантажувального USB-накопичувача для Windows

У BIOS ноутбука змінено пріоритет завантаження, після чого розпочато інсталяцію системи. У процесі встановлення обрано SSD накопичувач, як

окремий розділ під операційну систему та використано HDD для майбутніх даних сервера. Після завершення встановлення проведено стандартне налаштування параметрів користувача, оновлено драйвери, перевірено та встановлено всі доступні системні оновлення через Windows Update (рис. 3.4).

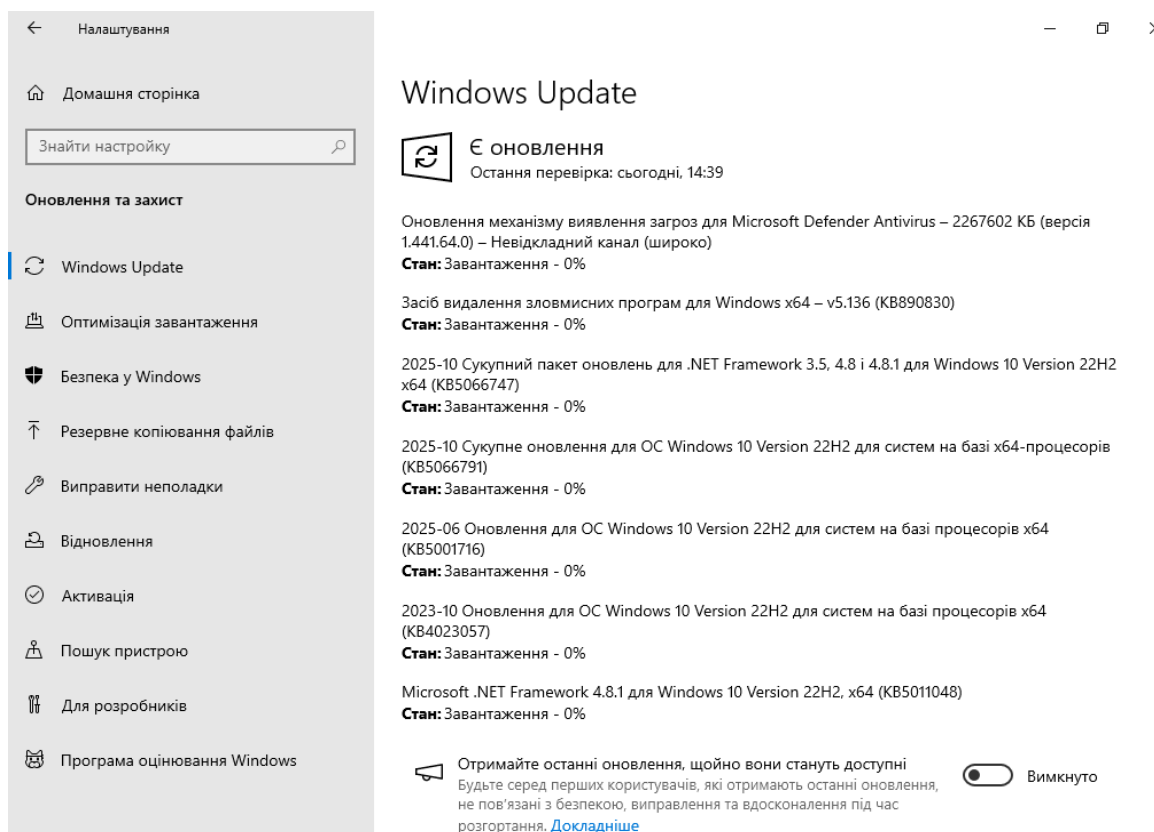


Рисунок 3.4 – Оновлення ОС

Для встановлення операційної системи Xubuntu також підготовлено завантажувальний USB-накопичувач. Для цього використано програму Rufus (рис. 3.5).

Перед встановленням виконано перевірку цілісності ISO-образу через контрольні суми SHA256, щоб уникнути помилок під час інсталяції. В процесі встановлення системи обраний системний розділ, swar-область і додатковий розділ для користувацьких даних.

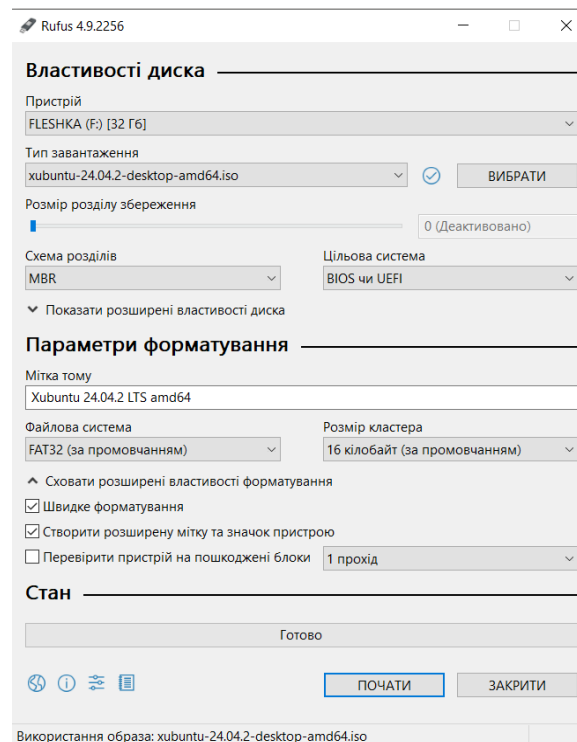


Рисунок 3.5 – Створення завантажувального USB-накопичувача для Xubuntu

Після завершення встановлення виконано оновлення системи через термінал командою `sudo apt update && sudo apt upgrade`, що дозволило отримати найновіші версії компонентів (рис. 3.6).

```
File Edit View Terminal Tabs Help
vboxuser@xubuntu:~$ sudo apt update && apt upgrade -y
[sudo] password for vboxuser:
Get:1 file:/cdrom noble InRelease
Ign:1 file:/cdrom noble InRelease
Get:2 file:/cdrom noble Release
Err:2 file:/cdrom noble Release
  File not found - /cdrom/dists/noble/Release (2: No such file or directory)
Hit:3 http://ua.archive.ubuntu.com/ubuntu noble InRelease
Get:4 http://ua.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:5 http://ua.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:6 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:7 http://ua.archive.ubuntu.com/ubuntu noble/main Translation-en [513 kB]
Get:8 http://ua.archive.ubuntu.com/ubuntu noble/restricted Translation-en [18.7
kB]
Get:9 http://ua.archive.ubuntu.com/ubuntu noble/universe Translation-en [5,982 k
B]
Get:10 http://ua.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118
kB]
Get:11 http://ua.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1,
585 kB]
Get:12 http://ua.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [29
9 kB]
Get:13 http://ua.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [
175 kB]
```

Рисунок 3.6 – Процес оновлення пакетів ОС

Операційна система Raspberry Pi OS встановлювалась на пристрій Raspberry Pi 5. Для цього використано офіційний інструмент Raspberry Pi Imager, за допомогою якого образ операційної системи записано на microSD-карту (рис. 3.7).

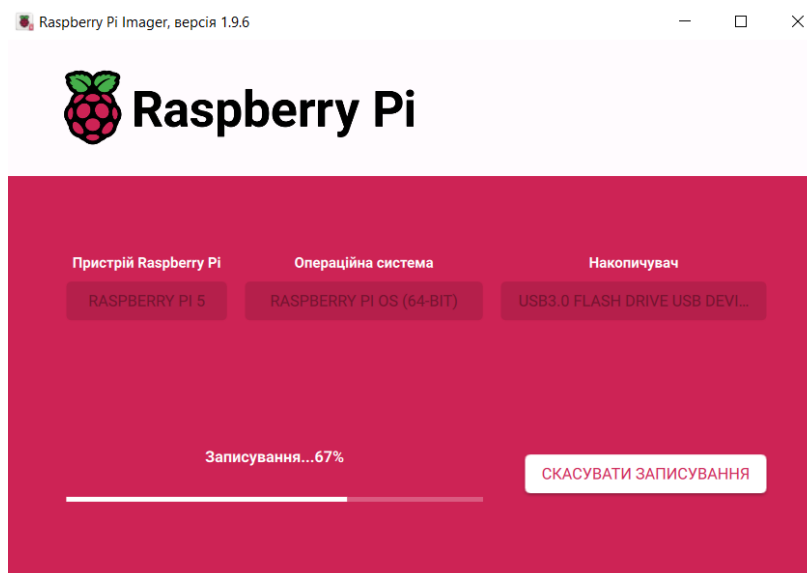


Рисунок 3.7 – Процес записування образу операційної системи на накопичувач

Після першого завантаження виконується оновлення системи (рис. 3.8) для отримання найновіших версій компонентів.

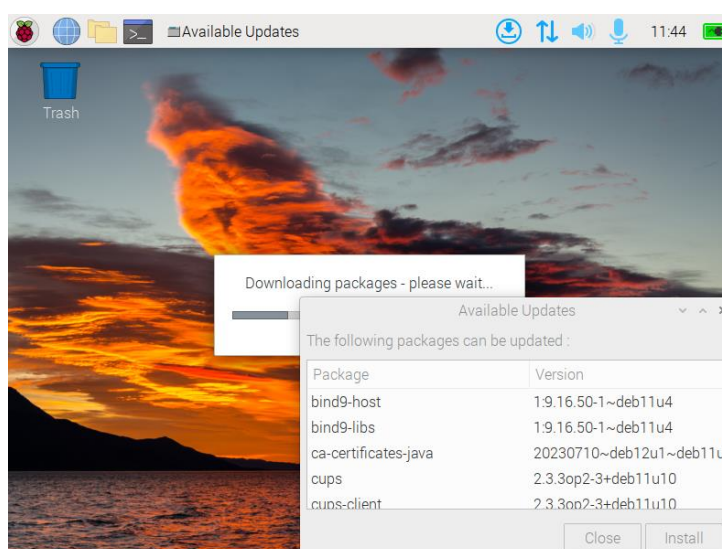
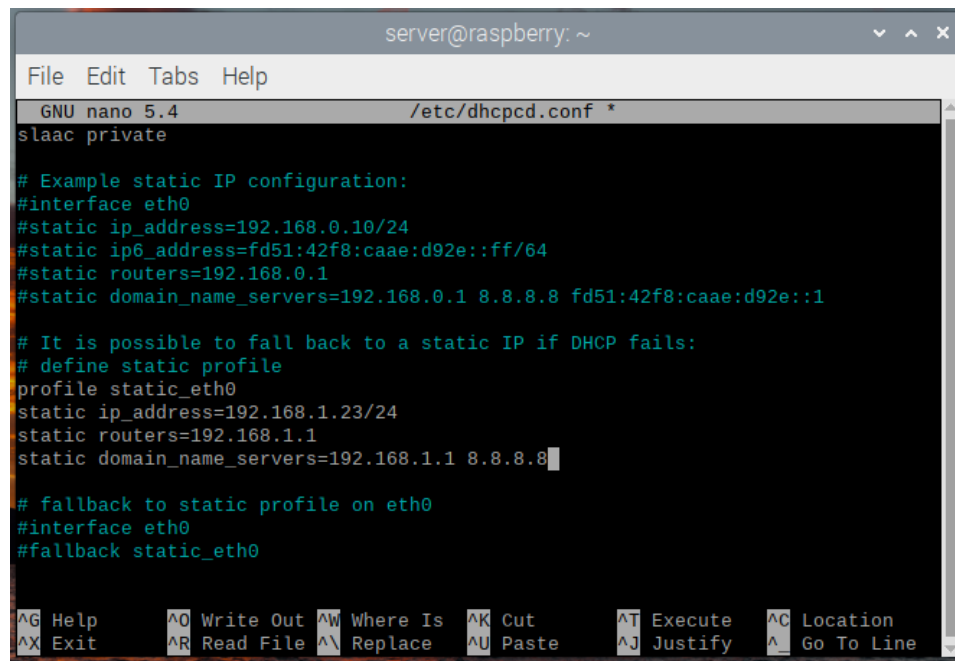


Рисунок 3.8 – Оновлення системи

Додатково налаштовано статичну IP-адресу (рис. 3.9), що дало можливість уникнути зміни адреси при перезавантаженні маршрутизатора.



```
server@raspberrypi: ~  
File Edit Tabs Help  
GNU nano 5.4 /etc/dhcpd.conf *  
slaac private  
  
# Example static IP configuration:  
#interface eth0  
#static ip_address=192.168.0.10/24  
#static ip6_address=fd51:42f8:caae:d92e::ff/64  
#static routers=192.168.0.1  
#static domain_name_servers=192.168.0.1 8.8.8.8 fd51:42f8:caae:d92e::1  
  
# It is possible to fall back to a static IP if DHCP fails:  
# define static profile  
profile static_eth0  
static ip_address=192.168.1.23/24  
static routers=192.168.1.1  
static domain_name_servers=192.168.1.1 8.8.8.8  
  
# fallback to static profile on eth0  
#interface eth0  
#fallback static_eth0  
  
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  ^C Location  
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify  ^_ Go To Line
```

Рисунок 3.9 – Налаштування статичної IP-адреси

Таким чином, етап первинного налаштування конфігурації дозволив створити повноцінні багатofункціональні серверні платформи. Кожна операційна система налаштована відповідно до своїх переваг та з метою оптимізації.

3.4 Модернізація ОС

В процесі підготовки серверного середовища здійснено оптимізацію та модернізацію системи Windows 10 з метою підвищення стабільності роботи, зменшення навантаження на апаратні ресурси та покращення енергоефективності. Оптимізаційні заходи спрямовані на видалення зайвого програмного забезпечення, налаштування енергоспоживання, керування системними службами та використання спеціалізованих інструментів.

В першу чергу здійснено видалення (рис. 3.10) програмного забезпечення яке не використовуватиметься в ході експлуатації пристрою та може знижувати

загальну ефективність пристрою. Дані дії виконуються за допомогою Панелі керувань.

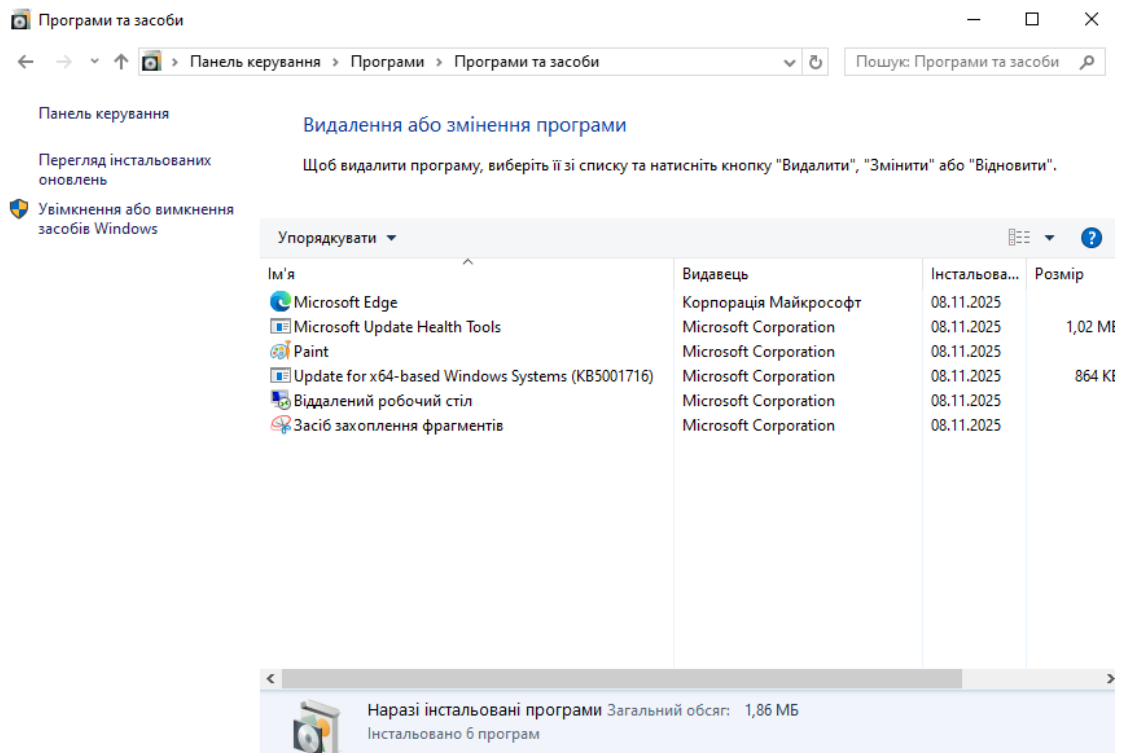


Рисунок 3.10 – Список програм, які залишилися після видалення

Під видалення потрапили демонстраційні додатки, стандартні утиліти, встановлені виробником, а також допоміжні компоненти, що не мають практичного значення для серверного середовища. Це дозволило зменшити кількість фонових процесів, знизити навантаження на оперативну пам'ять та звільнити місце на системному диску, що позитивно вплинуло на швидкодію комп'ютера.

Окрім цього змінено налаштування плану живлення. Повернувшись на головну сторінку панелі керувань та перейшовши в Електроживлення можна змінити або створити план живлення (рис. 3.11). В залежності від обраного плану змінюватись ефективність роботи та споживання енергії.

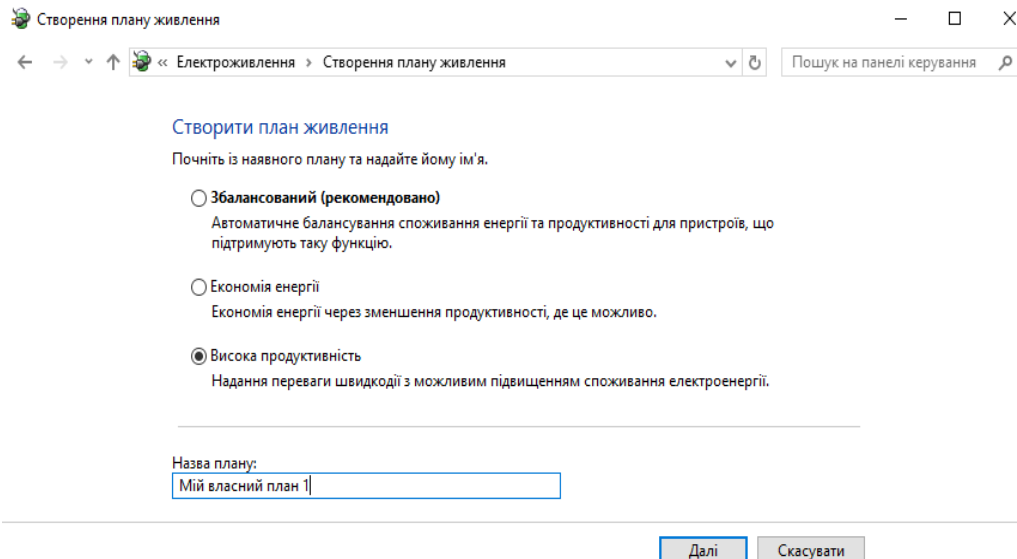


Рисунок 3.11 – Створення плану живлення

Ще одним не менш ефективним способом підвищити продуктивність є зниження графічного навантаження. Оскільки Windows 10 в більшості випадків використовує лише графічний інтерфейс, немає можливості вимкнути його і перейти в графічний. В зв'язку з цим проведено відключення візуальних ефектів інтерфейсу Windows (рис. 3.12).

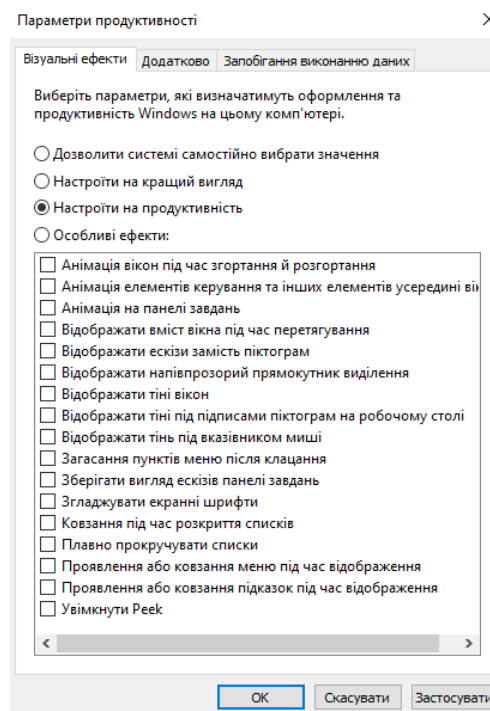


Рисунок 3.12 – Вимкнення візуальних ефектів

Зокрема вимкнено анімації, прозорість та додаткові елементи графічного оформлення. Для цього слід перейти в Панелі керувань в систему та обрати пункт Додаткові налаштування системи. Це дасть змогу зменшити використання ресурсів відеоадаптера і прискорити відгук системи на дії користувача, що є важливим аспектом для серверних конфігурацій, де пріоритетом є швидкодія.

Для пришвидшення процесу модернізації системи використано спеціальний PowerShell-скрипт `irm "https://christitus.com/win" | iex` (рис. 3.13). Завдяки даному скрипту запускається інструмент, що автоматизує процедуру оптимізації Windows 10 шляхом вимкнення непотрібних служб, телеметрії, елементів реклами. Завдяки чому підвищується продуктивність і зменшується навантаження на апаратні ресурси та систему.



Рисунок 3.13 – Інструмент для оптимізації роботи системи

- Connected User Experiences and Telemetry;
- Diagnostic Policy Service;
- Distributed Link Tracking Client;
- Program Compatibility Assistant Service;
- Secondary Logon;
- Offline Files;
- Smart Card;
- Superfetch (SysMain) ;
- Windows Insider Service;
- Remote Desktop Services;
- Remote Access Connection Manager;
- Windows Camera Frame Server;
- Windows Image Acquisition;
- Windows Biometric Service
- Sensor Service;
- Sensor Data Service;
- Sensor Monitoring Service;
- Touch Keyboard and Handwriting Panel Service;
- Phone Service;
- WalletService;
- Xbox Live Auth Manager;
- Xbox Live Game Save;
- Xbox Live Networking Service;
- Geolocation Service;
- Windows Mobile Hotspot Service;
- Windows Update Medic Service;
- Windows Media Player Network Sharing Service;
- Retail Demo Service;
- Security Center;

- Downloaded Maps Manager;
- AllJoyn Router Service;
- Windows Connect Now Config Registrar;
- Parental Controls;
- Wi-Fi Direct Services Connection Manager;
- Data Sharing Service;
- Data Collection Publishing Service;
- Diagnostic Execution Service;
- Diagnostic Service Host;
- Diagnostic System Host;
- Windows Push Notifications User Service;
- PrintWorkflow User Service;
- Application Layer Gateway Service;
- App Readiness;
- MapsBroker;
- Netlogon;
- Workstation;
- Windows Defender Advanced Threat Protection Service;
- Remote Access Auto Connection Manager;
- GameDVR and Broadcast User Service;
- Windows Perception Simulation Service;
- Windows Mixed Reality OpenXR Service;
- Tablet PC Input Service;
- Windows Connect Now;
- Windows Time;
- BitLocker Drive Encryption Service;
- CNG Key Isolation;
- Network Connection Sharing;
- Hyper-V Heartbeat Service;

- Hyper-V Remote Desktop Virtualization Service;
- Hyper-V Time Synchronization Service;
- Hyper-V Guest Shutdown Service;
- Hyper-V Data Exchange Service;
- Hyper-V Virtual Machine Session Service.

В результаті виконаних модернізаційних дій вдалося суттєво підвищити стабільність та швидкодію системи. Окрім цього, вдалося зменшити навантаження на оперативну пам'ять більш ніж на 10 % та значно скоротити число процесів і потоків (рис. 3.15).



Рисунок 3.15 – Зміни після модифікації ОС Windows 10

На відміну від багатьох сучасних операційних систем, Xubuntu не має великої кількості попередньо встановлених програм чи служб, які працюють у фоновому режимі. Всі сервіси є або системно необхідними, або належать до критично важливих компонентів, завдяки чому потреба в глобальній оптимізації мінімальна. Система створена на принципах продуктивності, стабільності та простоти. Найефективнішим кроком у модернізації Xubuntu є вимкнення графічного інтерфейсу. Графічне середовище постійно споживає

частину оперативної пам'яті та створює постійне навантаження на систему. У серверному застосуванні ці ресурси краще перенаправити на роботу сервісів.

Для вимкнення графічного інтерфейсу застосовується команда `sudo systemctl set-default multi-user.target`. Дана команда завантажує систему в текстовому режимі. У цьому режимі система завантажує лише необхідні служби, при цьому використання оперативної пам'яті може зменшитися майже удвічі, а навантаження на процесор стає стабільнішим. Керування системою здійснюється повністю через термінал або віддалено за допомогою SSH. В разі потреби графічний інтерфейс можна ввімкнути командою `sudo systemctl set-default graphical.target`. Завдяки вимкненню графічного інтерфейсу навантаження на систему значно зменшується (рис. 3.16). Фактично вимкнення графічного інтерфейсу в Xubuntu дає майже такий самий результат, як і від всіх модернізаційних дій на Windows 10. Xubuntu та Raspberry Pi OS працюють на базі Linux, тому процес оптимізації їхньої роботи ідентичний. Загалом завдяки здійсненню дій можна підвищити ефективність системи до 20 %.

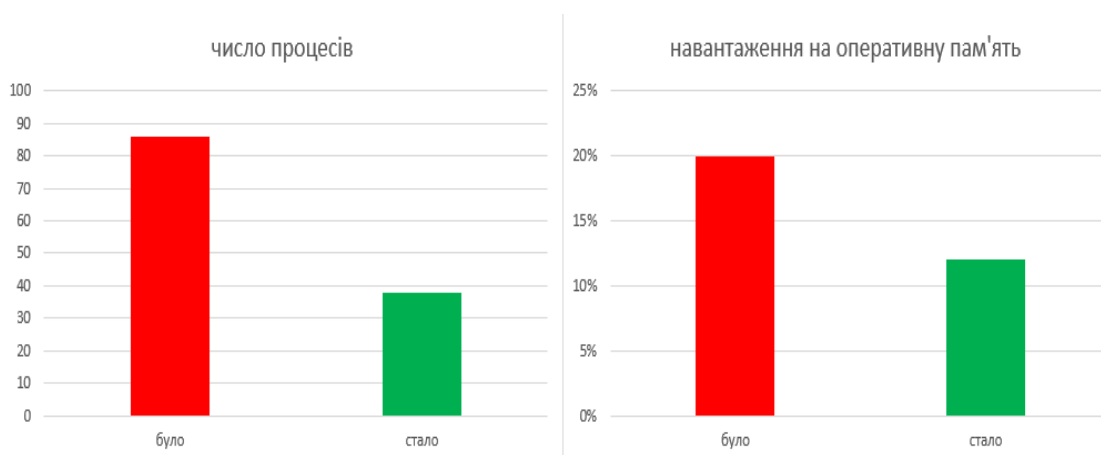


Рисунок 3.16 – Результати модернізації Xubuntu

3.5 Встановлення та налаштування сервісів

3.5.1 Медійний сервер

MiniDLNA, також відомий як ReadyMedia – це легкий DLNA/UPnP медіасервер, який дозволяє надавати доступ до медіафайлів (музика, відео,

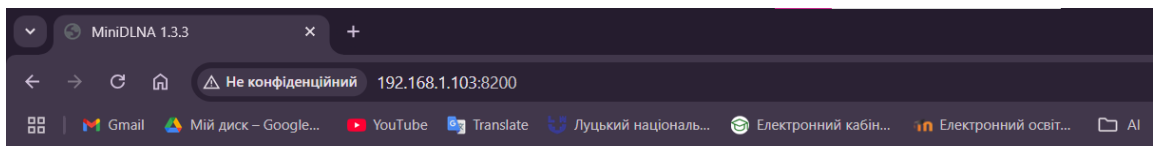
зображення) у локальній мережі. Він працює за протоколом DLNA/UPnP, що дозволяє сумісним пристроям, таким як телевізори, смартфони, медіаплеєри знаходити та відтворювати файли з сервера без потреби копіювання їх на кінцевий пристрій.

MiniDLNA в повній мірі реалізує підтримку протоколів DLNA/UPnP (Digital Living Network Alliance та Universal Plug and Play), що дозволяють клієнтським пристроям автоматично знаходити сервер в мережі та отримувати доступ до медіатеки. Завдяки цьому MiniDLNA забезпечує високу сумісність з більшістю сучасних пристроїв і не вимагає встановлення додаткового програмного забезпечення.

Найбільшою перевагою MiniDLNA є його невибагливість до апаратних ресурсів. Це робить його ідеальним варіантом у випадках, коли низьке енергоспоживання грає ключову роль, а сам пристрій має обмежений ресурс продуктивності. Окрім цього, такий медіасервер не має графічного інтерфейсу, а всі налаштування конфігурацій відбуваються через текстовий редактор.

Завдяки такому поєднанню простоти та надійності MiniDLNA дозволяє організувати функціональну систему медіастрімінгу в домашніх умовах з мінімальними вимогами до обладнання та високим рівнем сумісності з різноманітними пристроями.

Процес налаштування медіасервера розпочинається з інсталяції програмного пакета MiniDLNA за допомогою команди `sudo apt install minidlna`. Після встановлення необхідно відредагувати конфігураційний файл `/etc/minidlna.conf` та визначити шляхи до каталогів, що використовуватимуться для зберігання медіафайлів. Зміни конфігурації застосовуються шляхом виконання команд `sudo service minidlna restart` та `sudo service minidlna force-reload`, що забезпечує перезавантаження медіасервера та активацію оновлених параметрів [23]. Доступ до веб-інтерфейсу медіасервера здійснюється через веб-браузер за IP-адресою сервера з портом 8200, де відображається статистика медіафайлів та перелік підключених клієнтів (рис. 3.17).



MiniDLNA status

Media library

Audio files	0
Video files	8
Image files	50

Connected clients

ID	Type	IP Address	HW Address	Connections
0	Generic UPnP 1.0	192.168.1.6	F8:B4:6A:D4:FE:C3	0
1	Generic UPnP 1.0	192.168.1.2	0E:61:E0:18:FF:6D	1

1 connection currently open

Рисунок 3.17 – Веб-інтерфейс медіасервера

Для перегляду медіаконтенту з інших пристроїв необхідно використати медіаплеєр з підтримкою протоколів DLNA/UPnP (рис. 3.18). Варто зазначити, що перегляд медіаконтенту відбувається з мінімальною затримкою та без впливу на саму якість контенту. В першу чергу це завдяки тому, що транслявання контенту відбувається в межах однієї локальної мережі.

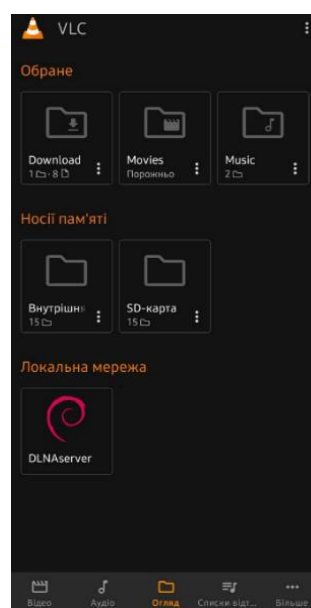


Рисунок 3.18 – Інтерфейс медіаплеєра з підтримкою DLNA/UPnP

У випадках, коли автоматичне оновлення медіатеки при додаванні нових файлів не відбувається, доцільно виконати примусову зупинку медіасервера, оновлення бази даних та повторний запуск за допомогою команд `sudo killall minidlnad`, `sudo minidlnad -R` та `sudo service minidlna start`.

Serviio – це безкоштовний багатофункціональний медіасервер, що дозволяє відтворювати медіаконтент на будь-якому пристрої, що сумісний з технологією DLNA в межах домашньої мережі.

Однією з головних переваг Serviio є широкий спектр форматів мультимедіа, які він підтримує, а також вбудованої системи транскодування, завдяки чому пристрої, що не підтримують певний формат файлу, все одно зможуть його відтворити. Оскільки Serviio в режимі реального часу перетворює формат файлу в сумісний, забезпечуючи безперервне відтворення. Така можливість робить Serviio більш універсальним рішенням у випадках, коли мультимедійна колекція містить файли різних типів.

Окрім цього Serviio підтримує роботу з онлайн-контентом. Користувач може додавати інтернет потоки, які будуть доступні одночасно з локальними файлами. Це сильно розширює можливості використання системи як єдиного медіацентру. Управління Serviio здійснюється через веб-інтерфейс, завдяки чому можна конфігурувати бібліотеки, доступ пристроїв, параметри трансляції тощо.

Таким чином, Serviio можна назвати більш просунутою альтернативою DLNA/UPnP. Окрім забезпечення легкого доступу до файлів медіаконтенту в локальній мережі, Serviio пропонує онлайн трансляцію, транскодування, підтримку великої кількості форматів та гнучке налаштування параметрів відтворення. Проте разом з цим всім Serviio менш енергоефективний та вимагає більше ресурсів.

Налаштування мультимедійного сервера Serviio в операційній системі Windows 10 починається з завантаження інсталяційного пакета з офіційного сайту розробника та подальшого його встановлення за допомогою стандартного

інсталятора Windows. Після завершення інсталяції запускається Serviio Console (рис. 3.19).

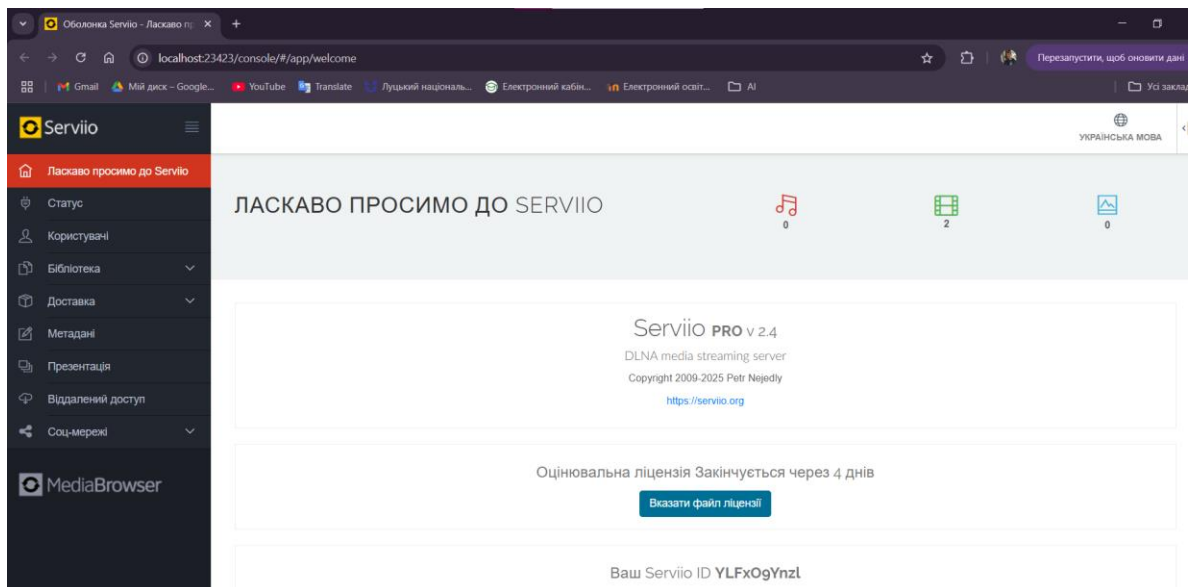


Рисунок 3.19 – Веб-інтерфейс Serviio

В розділі Бібліотека необхідно додати каталоги, що міститимуть мультимедійні файли шляхом натискання кнопки «Додати» та вибору відповідних тек. Для кожного каталогу вказується тип вмісту, що пришвидшить його індексацію та оптимізує подальше відтворення. В розділі статус у разі потреби можна вказати, які пристрої матимуть доступ, використовуючи фільтрацію за IP-адресами або мережевими інтерфейсами. Для забезпечення коректної роботи надається дозвіл Serviio у брандмауері Windows, що гарантує прийом вхідних з'єднань на відповідних портах. Після цього сервер автоматично починає індексацію доданих каталогів і стає доступним для клієнтських пристроїв, а саме телевізорів, персональних комп'ютерів та мобільних пристроїв, які мають встановлений медіаплеєр з підтримкою DLNA/UPnP. Для перевірки функціональності достатньо відкрити на сумісному пристрої медіаплеєр із підтримкою DLNA та обрати в списку доступних джерел Serviio Server, після чого можна здійснювати перегляд чи прослуховування мультимедійних файлів безпосередньо з сервера.

3.5.2 Ігровий сервер

Minecraft Server – це багатокористувацький ігровий сервер, який дозволяє гравцям грати разом в Minecraft в одній ігровій сесії. Розгортання ігрового сервера Minecraft починається з встановлення необхідного пакета Java, що забезпечує роботу сервера, командою `sudo apt install openjdk-21-jdk-headless`. Після чого створюється окремий каталог `mkdir minecraft` для розміщення файлів сервера та здійснюється перехід до нього командою `cd minecraft/`. Офіційний серверний файл завантажується з репозиторію Mojang за допомогою команди `wget`. Для первинного запуску сервера використовується команда `java -Xmx1024M -Xms1024M -jar server.jar nogui`, яка ініціює створення базових конфігураційних файлів. Після цього потрібно прийняти ліцензійну угоду, відкривши файл `eula.txt` в текстовому редакторі та встановивши параметр `eula=true` [24]. Повторний запуск сервера здійснюється тією ж командою і забезпечує його повне функціонування (рис. 3.20). У такому режимі сервер працює без графічного інтерфейсу, що знижує споживання ресурсів, роблячи його більш придатним для розгортання на пристроях з обмеженою обчислювальною потужністю. Для приєднання до ігрового сервера потрібно ввести IP-адресу сервера з портом 25565.

```
server@server-HP-250-G7-Notebook-PC:~/minecraft$ java -Xmx1024M -Xms1024M -jar server.jar nogui
Starting net.minecraft.server.Main
[12:03:25] [ServerMain/INFO]: Environment: Environment[sessionHost=https://sessionserver.mojang.com, service
sHost=https://api.minecraftservices.com, name=PROD]
[12:03:27] [ServerMain/INFO]: No existing world data, creating new world
[12:03:28] [ServerMain/INFO]: Loaded 1407 recipes
[12:03:28] [ServerMain/INFO]: Loaded 1520 advancements
[12:03:29] [Server thread/INFO]: Starting minecraft server version 1.21.8
[12:03:29] [Server thread/INFO]: Loading properties
[12:03:29] [Server thread/INFO]: Default game type: SURVIVAL
[12:03:29] [Server thread/INFO]: Generating keypair
[12:03:29] [Server thread/INFO]: Starting Minecraft server on *:25565
[12:03:29] [Server thread/INFO]: Using epoll channel type
[12:03:29] [Server thread/INFO]: Preparing level "world"
[12:03:41] [Server thread/INFO]: Preparing start region for dimension minecraft:overworld
[12:03:41] [Worker-Main-3/INFO]: Preparing spawn area: 2%
[12:03:42] [Worker-Main-3/INFO]: Preparing spawn area: 2%
[12:03:55] [Worker-Main-3/INFO]: Preparing spawn area: 7%
[12:04:24] [Server thread/INFO]: Time elapsed: 32272 ms
[12:04:24] [Server thread/INFO]: Done (59.379s)! For help, type "help"
[12:04:24] [Server thread/WARN]: Can't keep up! Is the server overloaded? Running 2008ms or 40 ticks behind
```

Рисунок 3.20 – Інтерфейс сервера

У випадку з Windows 10 відбуваються аналогічні налаштування. Проте для запуску сервера використовується файл з розширенням .bat, в який будуть записані наступні команди:

```
@ECHO OFF
```

```
java -Xms1024M -Xmx2048M -jar server.jar --nogui
```

```
pause
```

В результаті появиться можливість запускати сервер без потреби використання консолі.

3.5.3 Файлове сховище

Сервер Samba – це програмне забезпечення, що дозволяє Linux, Unix macOS та іншим операційним системам обмінюватись файлами, використовуючи протокол SMB/CIFS (Server Message Block / Common Internet File System). Завдяки цьому Samba забезпечує взаємодію між різними операційними системами. Основними його функціями є спільний доступ до файлів і принтерів та інтеграція з Active Directory, що дозволяє створити середовище для обміну файлами та використовувати єдину систему автентифікації для користувачів. Основні функції Samba полягають в наданні спільного доступу до файлів та принтерів у локальній мережі. Це дозволяє користувачам незалежно від операційної системи переглядати, копіювати, створювати та редагувати файли, що зберігаються на сервері. Дана можливість дозволяє використовувати пам'ять сервера як файлове сховище. Ще однією функцією Samba є Active Directory. Це означає, що можна налаштувати систему автентифікації та керування правами користувачів. Завдяки чому підтримується політика доступу та безпеки.

Налаштування файлового сервера на основі протоколу Samba починається з інсталяції відповідного пакета командою `sudo apt install samba`. Далі створюється каталог, який слугуватиме спільною директорією для обміну файлами між клієнтами. Окрім цього можна використовувати вже існуючі каталоги. Після цього необхідно відкрити конфігураційний файл `/etc/samba/smb.conf` у текстовому редакторі та додати параметри, що

визначатимуть спільний доступ, права користувачів та шлях до директорій. Для застосування змін виконується перезапуск служби за допомогою команди `sudo service smbd restart`. Додатково слід відкрити доступ через брандмауер для трафіку Samba, що реалізується командою `sudo ufw allow samba` [25]. В результаті проведених дій сервер стає доступним у локальній мережі, забезпечуючи обмін файлами між пристроями з підтримкою протоколу SMB/CIFS. Для доступу до файлового сервера необхідно ввести адресу сервера в файловий провідник (рис. 3.21).

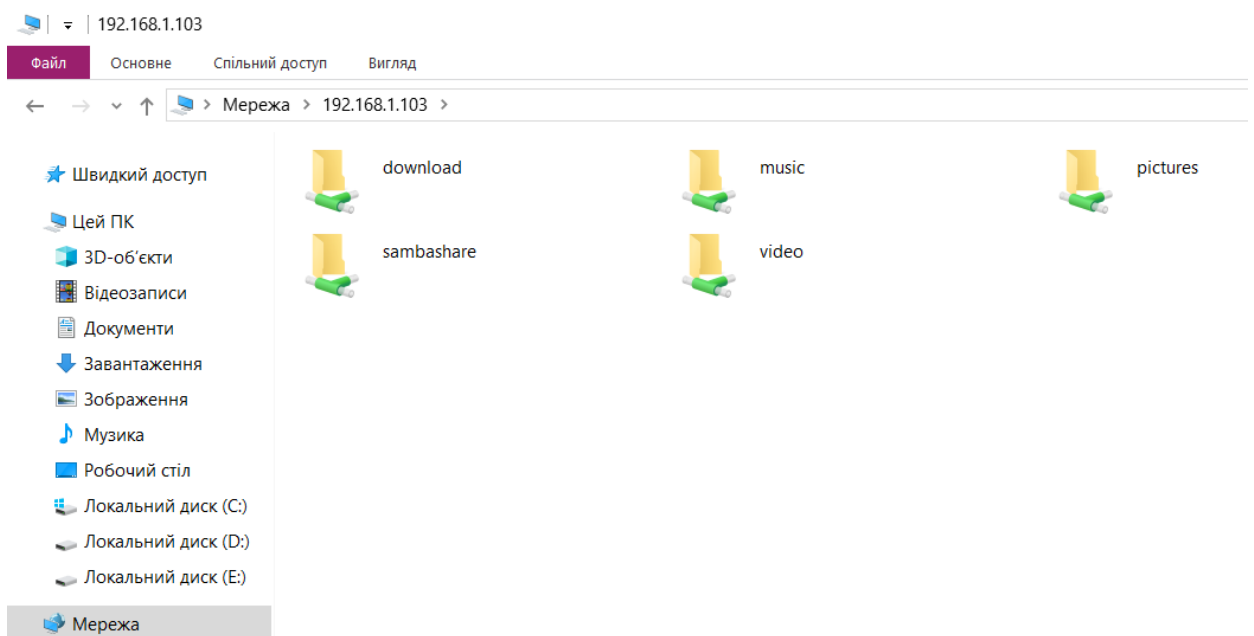


Рисунок 3.21 – Теки, до яких отримано доступ з іншого пристрою

Функція спільних тек у Windows 10 є одним з базових способів організації обміну файлів в середині локальної мережі. Вона реалізована за допомогою протоколу SMB (Server Message Block), що дозволяє комп'ютерам в мережі отримувати доступ до файлів, тек та принтерів. Налаштування спільної теки у Windows 10 не потребує встановлення додаткового програмного забезпечення, а саме налаштування відбувається за кілька хвилин. Для захисту даних передбачена система автентифікації користувачів та налаштування прав доступу. Для підключення до спільної теки користувачі повинні вести ім'я користувача та пароль, що належать до облікового запису на сервері. Це

забезпечує контроль над тим хто має доступ до файлів та обмежує користувачів в несанкціонованих діях. У разі необхідності можна створювати локальні облікові записи для різних користувачів і надавати їм різні права доступу. Таким чином спільна тека у Windows 10 так само, як і Samba, дозволяє використовувати пам'ять сервера як файлове сховище.

Налаштування спільного доступу до каталогу в операційній системі Windows 10 розпочинається з створення або вибору теки. Для цього у провіднику слід натиснути правою кнопкою миші по відповідній теці та обрати пункт властивості. У вікні властивостей необхідно перейти на вкладку «Спільний доступ». Після чого активувати параметр «Дозволити спільний доступ». Додатково можна налаштувати максимальну кількість одночасних підключень. Права доступу користувачів змінюються через вкладку «Безпека». Після збереження налаштувань варто перевірити коректність роботи «Служби спільного доступу», яка має бути увімкнена в параметрах мережі. У брандмауері Windows необхідно дозволити відповідний тип підключення. Для підключення до такої спільної теки з іншого пристрою використовується шлях у форматі \\IP-адреса_комп'ютера\Ім'я_теки (рис. 3.22).

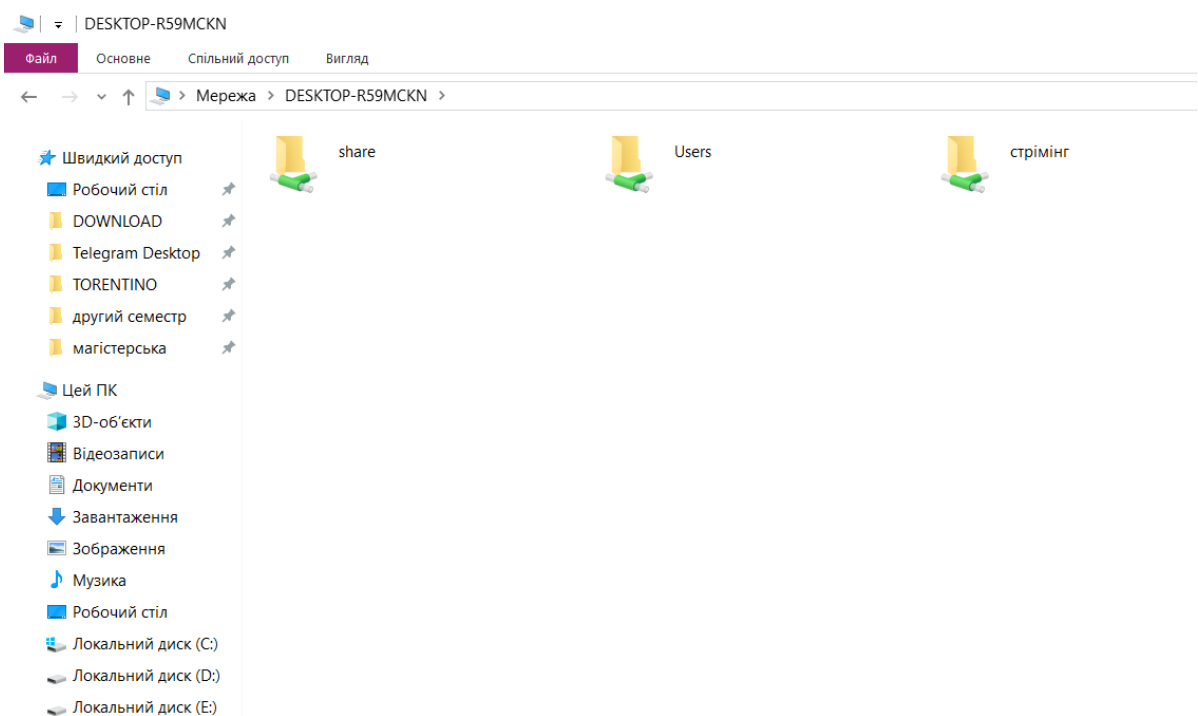


Рисунок 3.22 – Отриманий доступ до тек з іншого пристрою

3.6 Тестування та оцінка результатів

Для визначення ефективності роботи операційної системи Windows 10 як базової платформи домашнього сервера проведено серію досліджень. Основною метою цих досліджень після проведених модернізаційних заходів є оцінити навантаження на основі апаратні ресурси, енергоспоживання, температуру, швидкодію системи та час повного завантаження. Для Windows 10 та Xubuntu тестування проводилось на ноутбуці HP 250 G7 з такими характеристиками: процесор Intel Core i3-7020U (2 ядра, 2,3 ГГц), оперативна пам'ять 12 ГБ DDR4, накопичувач HDD 1 ТБ і SSD 256 ГБ, графічний процесор MDX110 та вбудований графічний процесор HD Graphics 620. У випадку з Raspberry Pi OS задіяний мікрокомп'ютер Raspberry Pi 5 з процесором Cortex-A76, 8 Гб оперативної пам'яті та 64 Гб вбудованої.

Основною метою тестування було визначити, наскільки стабільно кожна з конфігурацій може виконувати роль серверної системи в умовах різних рівнів навантаження від стану спокою до максимального використання ресурсів. Для цього обрані три режими дослідження:

- стан спокою. Система перебуває у ввімкненому стані, але сервіси не задіяні користувачем;
- стан середнього навантаження. Одночасно працює кілька служб, а передача даних відбувається періодично;
- стан максимального навантаження. Всі сервіси функціонують та використовуються в повній мірі. Відбувається обмін даними та активно використовується ігровий сервіс.

Дослідження систем відбувалось в однакових умовах, а саме: пристрої підключені до електромережі, використано 5 додаткових пристроїв, які виступали в ролі клієнта і користувались послугами сервера та усі пристрої під'єднані до однієї локальної мережі. Для отримання більш точних даних всі дослідження проводились по кілька разів.

Для забезпечення моніторингу споживання електроенергії використано EcoFlow, який дозволяв фіксувати в режимі реального часу показники потужності (Вт). Завдяки цьому виміряно реальне споживання електроенергії пристрою в різних режимах роботи. Для контролю параметрів системи використовувались диспетчери ресурсів. Окрім системних параметрів, додатково визначено час повного ввімкнення та завантаження системи. Середні показники записувались та оброблялись вручну. Додатково було зафіксовано дані до проведення модернізаційних заходів. Таким чином отримано достатню кількість даних до та після оптимізації і модернізації (рис. 3.23), що дозволяє здійснити порівняння, оцінити ефективність виконаних змін та визначити рівень підвищення продуктивності систем

		до			після				
Параметри		Windows 10 (HP 250 G7)	Xubuntu (HP 250 G7)	Raspberry Pi OS (Raspberrу Pi 5)	Параметри		Windows 10 (HP 250 G7)	Xubuntu (HP 250 G7)	Raspberry Pi OS (Raspberrу Pi 5)
Навантаження на ЦП/ОЗП	Мінімальне	8-20 % та 52 %	12-18 % та 20 %	2-12 % та 20 %	Навантаження на ЦП/ОЗП	Мінімальне	2-15 % та 38 %	8-12 % та 12 %	2-10 % та 12 %
	Середнє	13-50 % та 57 %	25-50 % та 37 %	18-53 % та 35 %		Середнє	7-40 % та 48 %	17-40 % та 24 %	15-50 % та 35 %
	Максимальне	40-97 % та 70 %	35-80 % та 43 %	42-85 % та 45 %		Максимальне	20-97 % та 52 %	30-80 % та 25 %	40-85 % та 45 %
Енергоспоживання	Мінімальне	20-27 Вт	15-23 Вт	4-6 Вт	Енергоспоживання	Мінімальне	16-25 Вт	15-20 Вт	4-5 Вт
	Середнє	25-40 Вт	17-30 Вт	7-9 Вт		Середнє	22-27 Вт	16-23 Вт	6-8 Вт
	Максимальне	53 Вт	37 Вт	10-13 Вт		Максимальне	30 Вт	25 Вт	10-12 Вт
Температура	Мінімальне	48°	40°	43°	Температура	Мінімальне	40°	38°	40°
	Середнє	65°	48°	57°		Середнє	53°	43°	55°
	Максимальне	80°	55°	70°		Максимальне	60°	50°	70°
Час ввімкнення та завантаження		2:40	2:25	1:12-1:40	Час ввімкнення та завантаження		0:40	0:25	1:12-1:40
Швидкість пересилання файлів		12Мб/с	11Мб/с	8-10Мб/с	Швидкість пересилання файлів		12Мб/с	11Мб/с	8-10Мб/с

Рисунок 3.23 – Дані отримані в ході дослідження

У процесі вимірювань спостерігалось, що в стані спокою система під керуванням Windows 10 споживає від 16 до 25 Вт при навантаженні на процесор у межах 2-15 % та 38 % використання оперативної пам'яті. При середньому навантаженні ці показники зростали до 22-27 Вт та 7-40 % навантаження ЦП, що є нормальним значенням для систем на базі Windows 10. У режимі максимального навантаження споживання сягало 30 Вт, а

навантаження процесора до 97 %, при цьому температура пристрою підвищувалась до 60° С. Отримані результати свідчать, що Windows 10 демонструє помірну стабільність і достатню продуктивність, однак споживання енергії є вищим порівняно з легшими системами на базі Linux. При цьому час завантаження системи становив у середньому 40 секунд, а швидкість передачі файлів досягала 12М б/с, що можна вважати прийнятним показником для домашнього середовища без спеціальної оптимізації мережевих параметрів.

У випадку з Xubuntu система створена на основі Ubuntu проте використовує робоче середовище XFCE, що відоме низьким використанням ресурсів та високою стабільністю. Це робить Xubuntu хорошим варіантом для використання як легке домашнє серверне середовище. Результати тестування показали нижчі показники навантаження та енергоспоживання ніж Windows 10. У стані спокою система споживає до 20 Вт при навантаженні на ЦП та ОЗУ 8-12 % та 12 % відповідно. При середньому навантаженні система видала споживання енергії в межах 16-23 Вт і навантаження на ЦП 17-40 %, що показує ефективне споживання ресурсів в даному режимі роботи. У піковому режимі система споживала до 25 Вт, при цьому навантаження процесора не перевищувало 80 %, а температура не перевищувала 50° С. Час повного завантаження системи становив у середньому 25 секунд, що дещо швидше, порівняно з Windows 10.

Під час тестування Raspberry Pi OS зафіксовано, що в стані спокою споживання енергії становило лише 4-5 Вт, при навантаженні на ЦП 2-10 % та використанні оперативної пам'яті до 12 %. У режимі середнього навантаження ці показники зростали до 6-8 Вт, а навантаження на процесор до 15-50 %, що вказує на високу енергоефективність пристрою навіть при активній роботі кількох сервісів. У стані максимального навантаження споживання не перевищувало 10-12 Вт, а температура процесора не піднімалась вище 70° С, що є хорошим показником для Raspberry Pi 5. Середній час повного завантаження системи складав від 1 хвилини 10 секунд до 1 хвилини 40 секунд.

3.7 Висновки з реалізації

Таким чином, проведене тестування підтвердило, що Windows 10 може використовуватись як основа для реалізації домашнього сервера. Разом з тим, її ефективність у плані енергоспоживання та часу реакції поступається більш легким дистрибутивам Linux. До плюсів можна віднести наявність готових рішень та інтуїтивність системи. Xubuntu забезпечує кращий баланс між продуктивністю та енергоспоживанням, що особливо важливо для систем, які працюють у режимі постійного ввімкнення. Висока стабільність, швидше завантаження та нижча температура компонентів роблять цей дистрибутив доцільним вибором для реалізації домашніх серверних рішень, таких як файлові або медіасервери. Результати експериментів показали, що Raspberry Pi OS вирізняється мінімальним енергоспоживанням, при цьому не такою стабільною поведінкою. Незважаючи на нижчу обчислювальну потужність у порівнянні з повноцінними ноутбуками, Raspberry Pi 5 забезпечує достатній рівень продуктивності для більшості типових домашніх серверних завдань.

Можна зробити висновок, що Windows 10 буде хорошим вибором для систем, які мають достатню кількість ресурсів і потребують швидких рішень та максимально стабільної поведінки. Xubuntu є оптимальним варіантом для створення енергоефективного, стабільного та невибагливого домашнього сервера, який забезпечує достатню продуктивність при відносно невеликих витратах електроенергії. Raspberry Pi OS є хорошим рішенням для створення компактного, безшумного та енергоефективного домашнього сервера, який не матиме серйозних навантажень на систему, проте може працювати цілодобово без суттєвого навантаження на електромережу.

ВИСНОВКИ

У ході дослідження проведено всебічний аналіз сучасних підходів до побудови та модернізації домашніх серверних систем. Вивчення існуючих рішень показало, що більшість користувачів прагнуть створити власну серверну інфраструктуру, проте вони зіштовхуються з проблемами обмеженої продуктивності, низької енергоефективності та складності налаштування. Це підтвердило актуальність теми роботи та необхідність розробки оптимізованого підходу до побудови стабільної і гнучкої домашньої серверної системи.

Основою дослідження стали аналіз та практична реалізація рішень, спрямованих на модернізацію домашньої серверної інфраструктури. В процесі дослідження проаналізовано існуючі рішення і підходи до організації домашніх серверів, визначено оптимальні інструменти для побудови системи, а також проведено практичну оптимізацію серверних середовищ.

В рамках реалізації проекту вибрано апаратне та програмне забезпечення для побудови серверної інфраструктури, в результаті чого здійснено розгортання домашнього сервера та налаштовано роботу ключових сервісів, а саме Samba, Minecraft server та MiniDLNA/Serviio. Реалізовано процес оптимізації та модернізації обраних систем, що дозволило зменшити енергоспоживання на 15% та підвищити стабільність роботи. Особливу увагу приділено оптимізації, завдяки чому досягнуто суттєвого зниження навантаження на процесор і оперативну пам'ять на 10% за рахунок вимкнення непотрібних графічних компонентів та фонових служб.

Проведено оцінку ефективності модернізації. Результати оцінки показали покращення загальної продуктивності системи та підвищення її стабільності при безперервній роботі. Оптимізація дозволила забезпечити стабільнішу роботу серверних сервісів, що підтверджує доцільність реалізованих рішень.

На основі отриманих результатів можна зробити висновок, що модернізація домашньої серверної інфраструктури є ефективним способом підвищення її надійності, енергоефективності та продуктивності. Розроблене

рішення може бути адаптоване для використання як у невеликих домашніх мережах так і в більш масштабних середовищах з розширеними вимогами до стабільності та автономності.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Багнюк Н. В., Бортник К. Я., С. Кайдик О. Л., Слюсар М. Я. Дослідження рішень та модернізація інфраструктури домашніх серверів. *Комп'ютерно-інтегровані технології: освіта, наука, виробництво*. Луцьк, 2025. № 60. С. 70-77.
2. Клієнт-серверна архітектура. Онлайн-курси від компанії QATestLab. Головна сторінка. URL: <https://training.qatestlab.com/blog/technical-articles/client-server-architecture/> (дата звернення: 05.08.2025).
3. Що таке сервер? Визначення, типи та особливості. ITEDU. IT Education Center Blog. URL: https://itedu.center/ua/blog/comparisons/types_of_servers/?srsltid=AfmBOoqF2cpbH6Pilkrl76nYGhrcDUi5L6pOe0fPgBVd5C2KDMBSP9Ch (дата звернення: 05.08.2025).
4. What Is Network Interface Card: Purposes, Functions & Types. *FS.com*. URL: https://www.fs.com/blog/what-is-a-network-interface-card-nic-definition-function-types-525.html?utm_source=chatgpt.com (date of access: 15.08.2025).
5. Linux. URL: <https://infdev.com.ua/docs/os/linux/> (date of access: 15.08.2025).
6. Ubuntu Server documentation. Ubuntu Server. URL: <https://documentation.ubuntu.com/server/> (date of access: 16.08.2025).
7. Windows documentation. Microsoft Learn: Build skills that open doors in your career. URL: <https://learn.microsoft.com/en-us/windows/> (date of access: 25.08.2025).
8. Raspberry Pi Documentation. Just a moment. URL: <https://www.raspberrypi.com/documentation/> (date of access: 07.09.2025).
9. Що таке сервер і як він працює: простими словами для бізнесу. Nixj. URL: <https://nixj.ua/scho-take-server--yak-vn-prasyu-prostimi-slovami-dlya-bznesu> (дата звернення: 19.09.2025).
10. Для чого потрібно резервне копіювання даних? Статті СЦ «Технарі». Сервісний центр – ремонт техніки в Києві. Технарі. URL:

<https://technari.com.ua/ua/services/about-company/articles/what-is-backup/> (дата звернення: 10.10.2025).

11. Що таке онлайн-інкрементальна резервна копія? Dropbox. Dropbox – 409. URL: https://www.dropbox.com/uk_UA/resources/incremental-backup (дата звернення: 10.10.2025).

12. Що таке Docker? Розгортання цієї популярної платформи додатків. DreamHost Blog. URL: <https://www.dreamhost.com/blog/uk/shcho-take-docker-rozgartannia-tsiieyi-populiarnoyi-platformi-dodatktiv-uk/> (дата звернення: 10.10.2025).

13. Підвищення безпеки сервера: що зробити насамперед. TutHost.ua. TutHost.ua. URL: <https://tuthost.ua/uk/blog/povyshenie-bezopasnosti-servera-hto-sdelat-v-pervuyu-ochered/> (дата звернення: 15.10.2025).

14. Zabbix 6.2 вже вийшов, і це його новини. Uibunlog. URL: <https://uk.uibunlog.com/zabbix-6-2-vzhe-vipuscheno,-i-ce-yogo-novini/> (дата звернення: 18.10.2025).

15. What Is NAS and How to Choose the Right Device for Your Needs?. Wondershare Recoverit. URL: <https://recoverit.wondershare.com/computer-tips/what-is-nas.html> (date of access: 23.10.2025).

16. NAS-сервер: що це, які задачі він вирішує і кому буде корисний? | Каталог цін E-Katalog. ек.ua – порівняння, відгуки, ціни в інтернет-магазинах. URL: https://ek.ua/ua/post/2750/285-nas-server-hto-eto-kakie-zadachi-reshaet-i-komu-budet-polezen/?&cgi_idsr=115761&gad_source=1&gad_campaignid=20042117003&gbraid=0AAAAAom0ad95DDXk-6DPIPXiKACLwXMWY&gclid=Cj0KCQjwjL3HBhCgARIsAPUg7a466bbG9TJdfUd7vB6EZvf7IUSAmMctk5ye8fS8YAK6it8HjsKUxXAaAsjLEALw_wcB (дата звернення: 27.10.2025).

17. Installing OpenMediaVault and setting up a NAS server. RecoverySoftware. URL: <https://recoverhdd.com/blog/installing-openmediavault-and-setting-up-a-nas-server.html?srsId=AfmBOoqH4DUa-OS9KUBKBxMo0qlKX8NvRSggSD20OnIYyRymyrhxQTMc> (date of access: 13.11.2025).

18. NextCloud для бізнесу: збереження та обмін даними. Техпідтримка.укр. URL: <https://xn--80ahcnklsjod4a9r.xn--j1amh/articles/nextcloud-yak-servis-dlya-zberezheniya-ta-obminu-danih-u-kompaniyi> (дата звернення: 13.11.2025).

19. Особливості файлових систем. Kingston Technology Company. URL: <https://www.kingston.com/ua/blog/personal-storage/understanding-file-systems> (дата звернення: 14.11.2025).

20. Типи мережевих протоколів і їх призначення (HTTP, IP, SSH, FTP, POP3, MAC). Оренда сервера. Віртуальні і виділені сервери в Україні, Європі, США. URL: https://deltahost.ua/ua/tipi-merezhevix-protokoliv-i-ih-priznachennya-http-ip-ssh-ftp-pop3-mac.html?srsltid=AfmBOoohzTQ84tGGiQQ1d_iOAgeCnbchF7XaMnyvGi4FLrZM-BG0wpAu (дата звернення: 15.11.2025).

21. Про Samba (Самба) в Ubuntu. Форум Ruby для початківців – вивчаємо Рубі разом!. URL: <https://rubydevelopers.org/t/samba-ubuntu/505> (дата звернення: 15.11.2025).

22. ReadyMedia – ArchWiki. ArchWiki. URL: <https://wiki.archlinux.org/title/ReadyMedia> (date of access: 15.11.2025).

23. MiniDLNA: A Beginner-Friendly Guide to Self-Hosting. Selfhostedninja. 04.11.2024. URL: https://www.selfhostedninja.com/minidlna-a-beginner-friendly-guide-to-self-hosting/?utm_source=chatgpt.com (дата звернення: 21.11.2025).

24. Setting up a Java Edition server. minecraft.wiki. URL: https://minecraft.wiki/w/Tutorial:Setting_up_a_Java_Edition_server (дата звернення: 27.11.2025).

25. Install and Configure Samba. Ubuntu. tutorials. URL: <https://ubuntu.com/tutorials/install-and-configure-samba#1-overview> (дата звернення: 27.11.2025).