

DOI: <https://doi.org/10.36910/6775-2524-0560-2025-60-07>

УДК 004.72

Багнюк Наталія Володимирівна, к.т.н., доцент<https://orcid.org/0000-0002-7120-5455>**Бортник Катерина Яківна**, к.т.н., доцент<https://orcid.org/0000-0001-5282-099X>**Кайдик Олег Леонтійович**, к.т.н., доцент<https://orcid.org/0000-0002-3620-270X>**Слюсар Максим Ярославович**, магістрант

Луцький національний технічний університет, м. Луцьк, Україна

ДОСЛІДЖЕННЯ РІШЕНЬ ТА МОДЕРНІЗАЦІЯ ДОМАШНЬОЇ СЕРВЕРНОЇ ІНФРАСТРУКТУРИ

Багнюк Н. В., Бортник К. Я., Кайдик О. Л., Слюсар М. Я. Дослідження рішень та модернізація домашньої серверної інфраструктури. У статті розглядаються можливості побудови та оптимізації домашньої серверної інфраструктури з використанням різних апаратно-програмних рішень. Проведено порівняльне дослідження трьох конфігурацій серверів на базі операційних систем Windows 10, Xubuntu та Raspberry Pi OS. Для забезпечення основних функцій сервера реалізовано медіастрімінг, ігровий сервер та файлове сховище. На платформі Xubuntu застосовано сервіси Samba (файлове сховище), MiniDLNA (медіастрімінг) та Minecraft Server (ігровий сервер). На Raspberry Pi OS використано аналогічний набір сервісів, адаптований під архітектуру ARM. Конфігурація на Windows 10 реалізована за допомогою Serviio (медіастрімінг), спільний доступ (файлове сховище) та Minecraft Server (ігровий сервер). Кожна з конфігурацій протестована у трьох режимах роботи: у стані спокою, при середньому та максимальному навантаженні. Для кожного режиму проведено вимірювання показників навантаження на процесор та оперативну пам'ять, енергоспоживання, часу ввімкнення та завантаження, температури та швидкості мережевого з'єднання. Результати вимірювань систематизовані та проаналізовані з метою визначення найбільш ефективного рішення для побудови домашнього сервера з урахуванням енергоефективності, стабільності роботи та швидкодії системи. Отримані результати можуть бути використані для вибору оптимальної конфігурації серверного обладнання та програмного забезпечення при розробці та модернізації домашніх серверних систем.

Ключові слова: домашній сервер, Windows 10, Xubuntu, Raspberry Pi OS, Samba, MiniDLNA, Minecraft Server, Serviio, медіастрімінг, файлове сховище, ігровий сервер.

Bahniuk N., Bortnyk K., Kaidyk O., Sliusar M. Researching solutions and modernizing home server infrastructure. The article considers the possibilities of building and optimizing a home server infrastructure using various hardware and software solutions. A comparative study of three server configurations based on the Windows 10, Xubuntu and Raspberry Pi OS operating systems was conducted. To ensure the main functions of the server, media streaming, game server and file storage were implemented. The Samba (file storage), MiniDLNA (media streaming) and Minecraft Server (game server) services were used on the Xubuntu platform. A similar set of services adapted to the ARM architecture was used on the Raspberry Pi OS. The configuration on Windows 10 was implemented using Serviio (media streaming), sharing (file storage) and Minecraft Server (game server). Each of the configurations was tested in three operating modes: idle, at average and maximum load. For each mode, the load on the processor and RAM, power consumption, startup and boot time, temperature and network connection speed were measured. The measurement results were systematized and analyzed to determine the most effective solution for building a home server, taking into account energy efficiency, stability and system performance. The results obtained can be used to select the optimal configuration of server hardware and software when developing and upgrading home server systems.

Keywords: home server, Windows 10, Xubuntu, Raspberry Pi OS, Samba, MiniDLNA, Minecraft Server, Serviio, media streaming, file storage, game server.

Постановка наукової проблеми. В час коли цифрові технології активно розвиваються, а кількість пристроїв, якими користується одна людина, росте, домашні сервери є важливою складовою сучасної цифрової інфраструктури. Такі сервери дозволяють централізувати зберігання та обробку даних, зменшити залежність від сторонніх хмарних сервісів і підвищити рівень приватності. Однак, на практиці створення домашнього сервера стикається з певною кількістю проблем: різні апаратні платформи та операційні системи мають значні відмінності у рівні ефективності, показниках надійності та напрямках практичного застосування. Відсутність чітких критеріїв вибору конфігурації призводить до того, що користувачі часто обирають не найбільш оптимальні рішення. Актуальним завданням є проведення комплексного дослідження продуктивності, енергоспоживання та надійності різних серверних рішень на базі популярних операційних систем та апаратних платформ з метою визначення найбільш ефективної моделі побудови домашнього сервера.

Аналіз останніх досліджень і публікацій. За останні роки збільшується кількість випадків використання персональних комп'ютерів та одноплатних мікрокомп'ютерів як основи для створення домашніх серверів.

У дослідженні [1] представлено процес побудови домашнього сервера на основі Raspberry Pi. Автор акцентує увагу на енергоефективності, надійності та компактності, вказуючи на забезпечення безперервної роботи системи.

У статті [2] розглянуто практичний підхід до перетворення звичайного ПК на повноцінний домашній сервер, з описом встановлення та налаштування сервісів. Автори наголошують на гнучкості такого рішення, оскільки воно дозволяє використати вже наявне обладнання та забезпечує масштабованість за рахунок зміни або розширення конфігурації.

Однак, у більшості робіт увага приділяється окремим аспектам або загальним порадам [3]. Комплексні дослідження, що поєднують порівняння різних платформ та операційних систем за показником продуктивності, енергоспоживання та стабільності роботи в різних режимах, є обмеженими. Саме ця прогалина зумовлює актуальність проведеного дослідження, у якому проаналізовано роботу домашніх серверів на основі Windows 10, Xubuntu, Raspberry Pi OS.

Виділення невирішених раніше частин загальної проблеми. Попри значну кількість робіт, присвячених налаштуванню та використанню домашніх серверів, окремі аспекти залишаються недостатньо висвітленими:

- відсутність комплексного підходу до порівняння продуктивності домашніх серверів на базі різних операційних систем із урахуванням однакових завдань;

- відсутність систематизованих даних щодо ефективності різних апаратно-програмних конфігурацій для виконання багатofункціональних завдань у єдиному порівняльному дослідженні.

Мета дослідження: провести комплексний аналіз та порівняння ефективності роботи домашніх серверів на базі різних операційних систем з метою визначення оптимальної апаратно-програмної конфігурації для побудови та модернізації домашньої серверної інфраструктури.

Завдання дослідження: виконати встановлення та налаштування домашнього сервера на базі операційних систем Windows 10, Xubuntu, Raspberry Pi OS для реалізації функцій медіастрімінгу, ігрового сервера та файлового сховища. Налаштувати та оптимізувати роботу сервісів Samba, MiniDLNA, Minecraft Server, Serviio та спільного доступу відповідно до обраних платформ. Провести тестування кожної конфігурації в різних режимах з вимірюванням показників. Порівняти отримані результати для визначення найбільш ефективної конфігурації. Сформулювати рекомендації щодо модернізації домашньої серверної інфраструктури з урахуванням енергоефективності, стабільності та продуктивності.

Основна частина дослідження. Основу дослідження складають компоненти, які дають можливість використовувати відкрите програмне забезпечення, що створює умови для гнучкого налаштування та масштабування під конкретні потреби користувача.

Апаратною складовою є ноутбук HP 250 G7 з процесором Intel Core I3 7th Gen, графічним процесором MDX110, 12 Гб оперативної пам'яті та 1 Тб вбудованої. Окрім цього, задіяний мікрокомп'ютер Raspberry Pi 5 з процесором Cortex-A76, 8 Гб оперативної пам'яті та 64 Гб вбудованої.

Для роботи домашнього сервера обрані операційні системи Windows 10, Xubuntu та Raspberry Pi OS, оскільки вони є одними з найпоширеніших представників операційних систем. Окрім цього вони пропонують широкий спектр можливостей, що дають змогу обрати оптимальну конфігурацію.

У випадку з операційними системами Xubutu та Raspberry Pi OS для забезпечення основних функцій сервера реалізовано медіастрімінг, ігровий сервер та файлове сховище, а саме, застосовано сервіси MiniDLNA, Minecraft Server та Samba відповідно.

MiniDLNA, також відомий як ReadyMedia – це легкий DLNA/UPnP медіасервер, який дозволяє надавати доступ до медіафайлів (музика, відео, зображення) по мережі. Він працює за протоколом DLNA/UPnP, що дозволяє сумісним пристроям, таким як телевізори, смартфони, медіаплеєри, знаходити та відтворювати файли з сервера.

Процес налаштування медіасервера розпочинається з інсталяції [4] 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. 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.1).

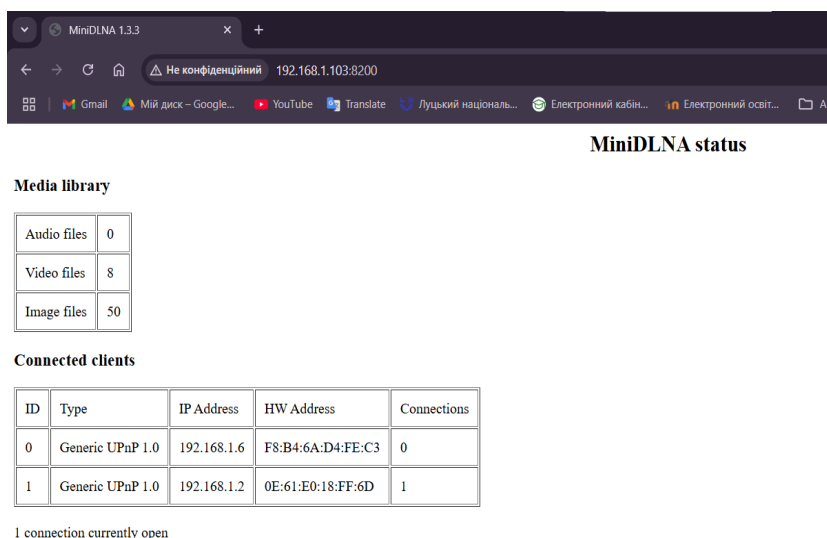


Рис.1. Веб-інтерфейс медіасервера

Для перегляду медіаконтенту з інших пристроїв необхідно використати медіаплеєр з підтримкою протоколів DLNA/UPnP (рис. 2).

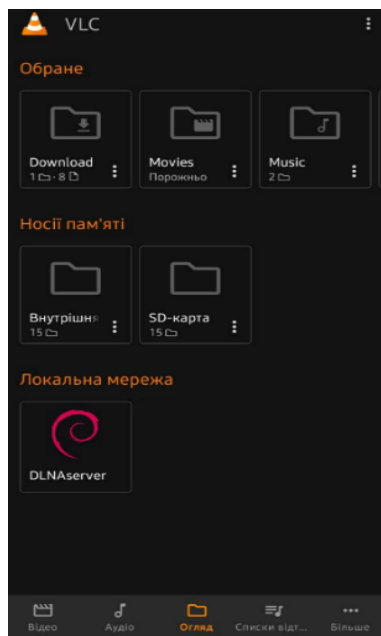


Рис. 2. Інтерфейс медіаплеєра з підтримкою DLNA/UPnP

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`.

Minecraft Server – це багатокористувацький ігровий сервер, який дозволяє гравцям грати разом в Minecraft в одній ігровій сесії [5]. 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 https://piston-data.mojang.com/v1/objects/6bce4ef400e4efaa63a13d5e6f6b500be969ef81/server.jar`. 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`. Повторний запуск сервера здійснюється тією ж командою і забезпечує його повне функціонування (рис. 3). 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
```

Рис. 3. Інтерфейс сервера

Сервер Samba – це програмне забезпечення, що дозволяє Linux, Unix macOS та іншим операційним системам обмінюватись файлами, використовуючи протокол SMB/CIFS. Основними його функціями є спільний доступ до файлів і принтерів та інтеграція з Active Directory, що дозволяє вільно обмінюватись файлами та використовувати єдину систему автентифікації для користувачів.

Налаштування файлового сервера на основі протоколу Samba починається з інсталяції відповідного пакета командою `sudo apt install samba` [6]. 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`. 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. 4).

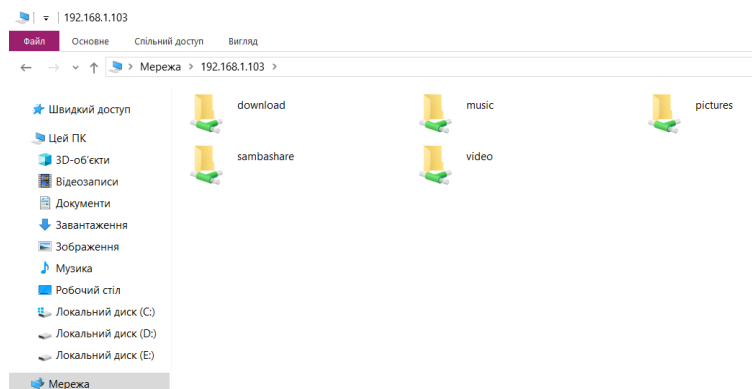


Рис. 4. Теки, до яких отримано доступ з іншого пристрою

На Windows 10 також можна реалізувати функції медіастрімінгу, ігрового сервера та файлового сховища. Проте це робиться за допомогою програм-аналогів або інструментів, вбудованих в операційну систему. Тому для забезпечення роботи медіастрімінгу обрано сервіс Serviio, ігровий сервер – Minecraft server, а функцію файлового сховища виконуватиме спільний доступ до теки. Окрім цього, Windows 10 дозволяє легко та інтуїтивно використовувати графічний інтерфейс.

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

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. 5).

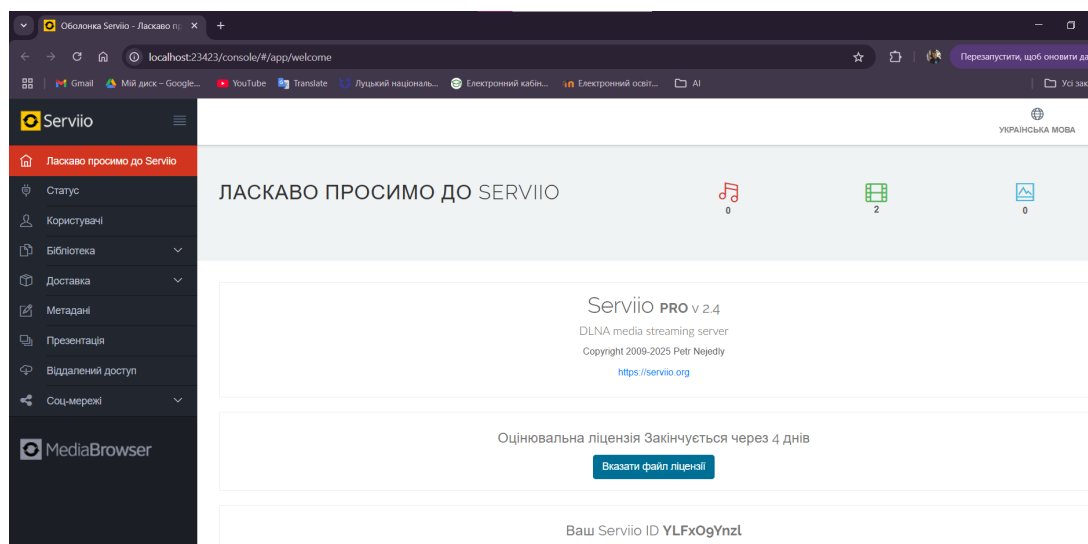


Рис. 5. Веб-інтерфейс Serviio

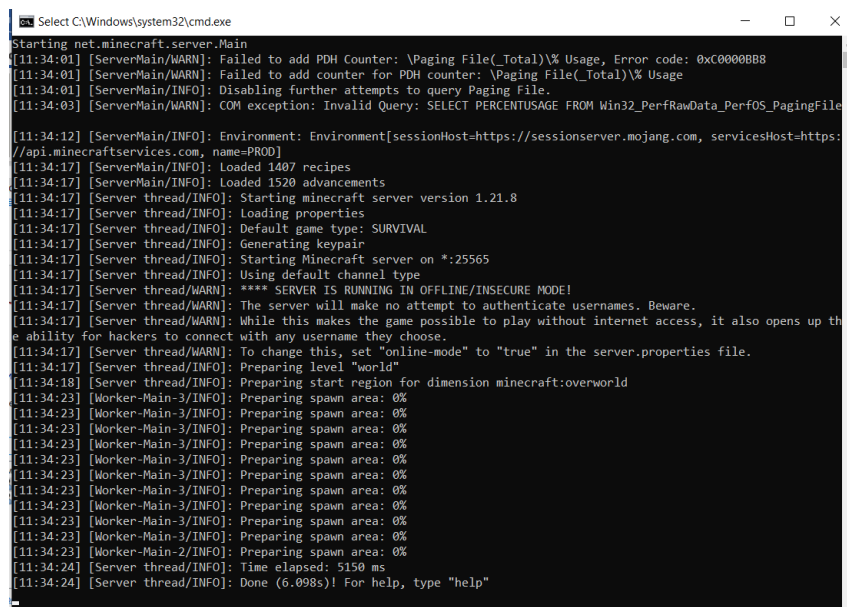
В розділі Бібліотека додаються каталоги, що міститимуть мультимедійні файли шляхом натискання кнопки Додати та вибору відповідних тек. Для кожного каталогу можна вказати тип вмісту, що пришвидшить його індексацію та оптимізує подальше відтворення. В розділі Статус у разі потреби можна вказати, які пристрої матимуть доступ, використовуючи фільтрацію за IP-адресами або мережевими інтерфейсами. Для забезпечення коректної роботи необхідно надати 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.

Налаштування і запуск Minecraft server на Windows 10 розпочинається з встановлення Java. Для цього необхідно завантажити актуальну версію Java Development Kit з офіційного сайту Oracle або OpenJDK, щоб забезпечити роботу серверного програмного забезпечення. Після встановлення створюється окрема робоча директорія, у якій зберігатимуться файли сервера. В цю директорію завантажується офіційний серверний файл з репозиторію Mojang. Для запуску файлу сервера створюється файл з розширенням .bat, в якому прописується наступна команда:

```
@ECHO OFF
java -Xms1024M -Xmx2048M -jar server.jar --nogui
pause
```

В результаті виконання цього файлу буде запускатись сервер. При першому запуску сервера слід прийняти ліцензійну угоду. Відкривши файл eula.txt в текстовому редакторі та встановивши параметр eula=true. Повторний запуск сервера здійснюється в той же спосіб через файл з

розширенням .bat. У такому режимі сервер працює без графічного інтерфейсу (рис. 6). Для приєднання до ігрового сервера необхідно ввести IP-адресу сервера з портом 25565.



```
Starting net.minecraft.server.Main
[11:34:01] [ServerMain/WARN]: Failed to add PDH Counter: \Paging File(_Total)\% Usage, Error code: 0xC00000B8
[11:34:01] [ServerMain/WARN]: Failed to add counter for PDH counter: \Paging File(_Total)\% Usage
[11:34:01] [ServerMain/INFO]: Disabling further attempts to query Paging File.
[11:34:03] [ServerMain/WARN]: COM exception: Invalid Query: SELECT PERCENTUSAGE FROM Win32_PerfRawData_PerfOS_PagingFile

[11:34:12] [ServerMain/INFO]: Environment: Environment[sessionHost=https://sessionserver.mojang.com, servicesHost=https://api.minecraftservices.com, name=PROD]
[11:34:17] [ServerMain/INFO]: Loaded 1407 recipes
[11:34:17] [ServerMain/INFO]: Loaded 1520 advancements
[11:34:17] [Server thread/INFO]: Starting minecraft server version 1.21.8
[11:34:17] [Server thread/INFO]: Loading properties
[11:34:17] [Server thread/INFO]: Default game type: SURVIVAL
[11:34:17] [Server thread/INFO]: Generating keypair
[11:34:17] [Server thread/INFO]: Starting Minecraft server on *:25565
[11:34:17] [Server thread/INFO]: Using default channel type
[11:34:17] [Server thread/WARN]: *** SERVER IS RUNNING IN OFFLINE/INSECURE MODE!
[11:34:17] [Server thread/WARN]: The server will make no attempt to authenticate usernames. Beware.
[11:34:17] [Server thread/WARN]: While this makes the game possible to play without internet access, it also opens up th
e ability for hackers to connect with any username they choose.
[11:34:17] [Server thread/WARN]: To change this, set "online-mode" to "true" in the server.properties file.
[11:34:17] [Server thread/INFO]: Preparing level "world"
[11:34:18] [Server thread/INFO]: Preparing start region for dimension minecraft:overworld
[11:34:23] [Worker-Main-3/INFO]: Preparing spawn area: 0%
[11:34:23] [Worker-Main-3/INFO]: Preparing spawn area: 0%
[11:34:23] [Worker-Main-3/INFO]: Preparing spawn area: 0%
[11:34:23] [Worker-Main-3/INFO]: Preparing spawn area: 0%
[11:34:23] [Worker-Main-3/INFO]: Preparing spawn area: 0%
[11:34:23] [Worker-Main-3/INFO]: Preparing spawn area: 0%
[11:34:23] [Worker-Main-3/INFO]: Preparing spawn area: 0%
[11:34:23] [Worker-Main-3/INFO]: Preparing spawn area: 0%
[11:34:23] [Worker-Main-2/INFO]: Preparing spawn area: 0%
[11:34:24] [Server thread/INFO]: Time elapsed: 5150 ms
[11:34:24] [Server thread/INFO]: Done (6.098s)! For help, type "help"
```

Рис. 6. Запущений ігровий сервер

Спільна тека в Windows 10 – це функція, яка дозволяє обмінюватись файлами та папками з іншими комп'ютерами в мережі. Це зручний спосіб для роботи над спільними проектами, доступу до файлів з інших пристроїв або обміну інформацією.

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. 7).

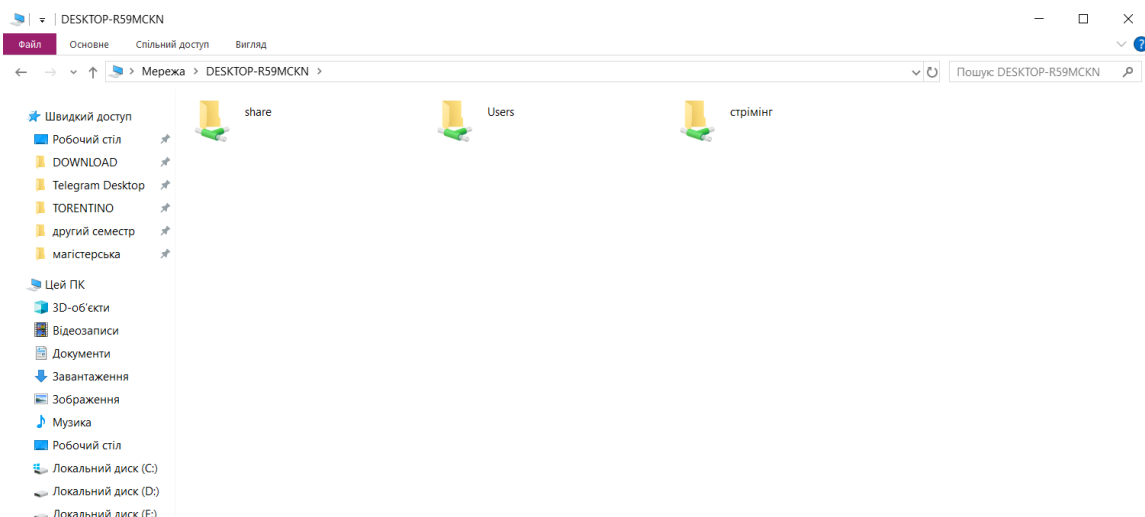


Рис. 7. Отриманий доступ до тек з іншого пристрою

Для проведення тестування ефективності роботи домашнього сервера використано комбінований підхід до вимірювання параметрів. Для визначення енергоспоживання

застосовувався прилад для заміру, який показував споживання ватт(Вт) в моменті. Для моніторингу завантаження центрального процесора, оперативної пам'яті та температурних показників використовувався диспетчер ресурсів. Такий підхід дав змогу отримати комплексну та об'єктивну картину ефективності роботи системи, а також порівняти показники у різних сценаріях використання.

В результаті проведених досліджень аналіз отриманих результатів тестування показує, що кожна з протестованих систем має свої переваги та не недоліки, а самі результати занесені в таблицю 1. Окрім цього, кожна система по своєму ефективна для різних потреб.

Таблиця 1. Отримані параметри при дослідженні

Параметри		Windows 10 (HP 250 G7)	Xubuntu (HP 250 G7)	Raspberry Pi OS (Raspberry Pi 5)
Навантаження на ЦП/ОЗП	Стан спокою	2-15 % та 38 %	8-12 % та 12 %	2-10 % та 20 %
	Середнє навантаження	7-40 % та 48 %	17-40 % та 24 %	15-50 % та 35 %
	Максимальне навантаження	20-97 % та 52 %	30-80 % та 25 %	40-85 % та 45 %
Енергоспоживання	Стан спокою	16-25 Вт	15-20 Вт	4-5 Вт
	Середнє навантаження	22-27 Вт	16-23 Вт	6-8 Вт
	Максимальне навантаження	30 Вт	25 Вт	10-12 Вт
Температура	Стан спокою	40°	38°	40°
	Середнє навантаження	53°	43°	55°
	Максимальне навантаження	60°	50°	70°
Час ввімкнення та завантаження (хв)		2,40	2,25	1,12-1,40
Швидкість пересилання файлів		12Мб/с	11Мб/с	8-10Мб/с

Windows 10 продемонструвала стабільно високу швидкість роботи та стійкість до відмов. Однак енергоспоживання є найбільшим серед протестованих конфігурацій. Навіть в стані спокою система споживає 16 Вт, а під максимальним навантаженням показник сягає 30 Вт. Навантаження на ЦП було нестабільним та варіювалось в досить великому діапазоні. При цьому температура залишалась в межах безпечних значень, хоч і була трохи вищою за інші системи.

Xubuntu показав кращу оптимізацію використання апаратних ресурсів. Навантаження на оперативну пам'ять навіть в пікових умовах не перевищувало 25 %, а енергоспоживання залишалось нижчим ніж у Windows 10. Температурний режим також виявився більш стабільним, що свідчить про ефективне керування апаратною частиною та відсутність зайвого навантаження у фонових процесах.

Raspberry Pi OS на базі Raspberry Pi 5 є лідером по енергоефективності. Навіть під час максимального навантаження система споживала не більше 12 Вт, що робить її ідеальною для безперервної роботи та використання у проєктах, де критичною є економія енергії. Водночас слід врахувати, що швидкість передачі даних та обчислювальні потужності були нижчими. Разом з цим температура під навантаженням досягала 70°. Відповідно до цього система потребує належного охолодження.

Таким чином, вибір оптимальної конфігурації для домашнього сервера безпосередньо залежить від вимог користувача. Якщо пріоритетом є продуктивність та швидкість роботи з великими обсягами даних, то краще буде використовувати Windows або Linux на потужному обладнанні. Якщо ж головним фактором є енергоефективність та компактність, то Raspberry Pi є оптимальним рішенням.

Висновки та перспективи подальшого дослідження. Домашній сервер є перспективною платформою для реалізації широкого спектру завдань. Його перевага полягає в можливості гнучкої адаптації під потреби користувача. Залежно від вимог, систему можна масштабувати, замінювати окремі компоненти або навіть об'єднувати декілька серверів в кластери для обробки великих обсягів даних. Такий підхід дозволяє розпочати з базової конфігурації для домашніх потреб і поступово розширювати функціонал в залежності від необхідності та наявних можливостей. Сучасні технології роблять домашні сервери ще більш універсальними. Наприклад, інтеграція штучного інтелекту або інтелектуальних систем управління дозволяє автоматизувати адміністрування та підвищити ефективність використання ресурсів. Такий домашній сервер може не лише виконувати заздалегідь налаштовані завдання, але й активно реагувати на зміну умов, оптимізуючи свою роботу. Це відкриває можливість створення високопродуктивних і водночас економічних рішень, здатних задовольнити як побутові так і напівпрофесійні чи професійні потреби.

Список бібліографічного опису:

1. Висоцький В. Домашній мультимедійний сервер на базі Raspberry PI / Тернопільський національний технічний університет імені Івана Пулюя. URL: https://elartu.tntu.edu.ua/bitstream/lib/42279/2/Vadim_Vysotsky.pdf (дата звернення: 03.06.2025).
2. Як перетворити будь-який ПК на домашній сервер. Sharpcoderblog. URL: <https://uk.sharpocoderblog.com/blog/how-to-turn-any-pc-into-a-home-server> (дата звернення: 21.06.2025).
3. Howik. Building a Home Server: The Ultimate Guide for 2025. howik. 25.05.2025. URL: <https://howik.com/building-a-home-server-guide> (дата звернення: 23.06.2025).
4. 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: 07.07.2025).
5. Setting up a Java Edition server. minecraft.wiki. URL: https://minecraft.wiki/w/Tutorial:Setting_up_a_Java_Edition_server (дата звернення: 14.07.2025).
6. Install and Configure Samba. Ubuntu. tutorials. URL: <https://ubuntu.com/tutorials/install-and-configure-samba#1-overview> (дата звернення: 16.07.2025).
7. Yin, Z., Zhang, M., & Kawahara, D. (2024). Harmony: A Home Agent for Responsive Management and Action Optimization with a Locally Deployed Large Language Model. arXiv preprint arXiv:2410.14252.

References:

1. Vysotsky V. Home multimedia server based on Raspberry PI / Ivan Pulyuy Ternopil National Technical University. URL: https://elartu.tntu.edu.ua/bitstream/lib/42279/2/Vadim_Vysotsky.pdf (access date: 03.06.2025).
2. How to turn any PC into a home server. Sharpcoderblog. URL: <https://uk.sharpocoderblog.com/blog/how-to-turn-any-pc-into-a-home-server> (access date: 21.06.2025).
3. Howik. Building a Home Server: The Ultimate Guide for 2025. howik. 25.05.2025. URL: <https://howik.com/building-a-home-server-guide> (дата звернення: 23.06.2025).
4. 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: 07.07.2025).
5. Setting up a Java Edition server. minecraft.wiki. URL: https://minecraft.wiki/w/Tutorial:Setting_up_a_Java_Edition_server (дата звернення: 14.07.2025).
6. Install and Configure Samba. Ubuntu. tutorials. URL: <https://ubuntu.com/tutorials/install-and-configure-samba#1-overview> (дата звернення: 16.07.2025).
7. Yin, Z., Zhang, M., & Kawahara, D. (2024). Harmony: A Home Agent for Responsive Management and Action Optimization with a Locally Deployed Large Language Model. arXiv preprint arXiv:2410.14252.