

Міністерство освіти і науки України
Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки

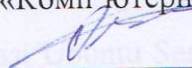
ПОЯСНЮВАЛЬНА ЗАПИСКА

до випускної кваліфікаційної роботи бакалавра
на тему: «Проект файлового сервера змішаного доступу
SMB/NFS на базі Linux»

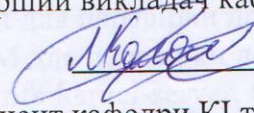
- операційна система Ubuntu Server для розгортання файлового сервера;
- протоколи мережевого файлового доступу SMB/CIFS та NFS;
- програмне забезпечення Samba для забезпечення доступу Windows-клієнтів;
- служба NFS Server для забезпечення доступу Linux-клієнтів;
- механізми POSIX-прав та POSIX-ACL для розмежування доступу;
- віртуалізоване середовище Oracle VirtualBox;
- клієнтські операційні системи Windows та Ubuntu Desktop;
- інструменти тестування Iperf3, nmap, smbclient, smbstatus.

Виконав: студент 4-го курсу групи КІ-22-1

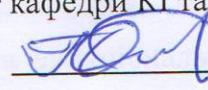
спеціальності 123 «Комп'ютерна інженерія»

 Максим ОСЕЦЬКИЙ

Керівник: старший викладач кафедри КІ та КБ

 Марія КОЛОЩУК

Рецензент: доцент кафедри КІ та КБ к.пед.н, доцент

 Олена ГОЛОВНЯ

Житомир - 2026

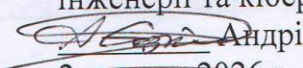
Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки
Спеціальність 123 «Комп'ютерна інженерія»
Освітня програма «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач

кафедри комп'ютерної

інженерії та кібербезпеки

 Андрій Єфіменко

2 лютого 2026 р.

ЗАВДАННЯ

на випускн у кваліфікаційну роботу бакалавра

Студент: Осецький Максим Петрович

Тема роботи: «Проект файлового сервера змішаного доступу SMB/NFS на базі Linux»

затверджена Наказом університету від 2 лютого 2026 р. № 57/с

Термін здачі студентом закінченої роботи 23 червня 2026 р.

Вихідні дані роботи:

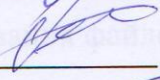
- операційна система Ubuntu Server для розгортання файлового сервера;
- протоколи мережевого файлового доступу SMB/CIFS та NFS;
- програмне забезпечення Samba для забезпечення доступу Windows-клієнтів;
- служба NFS Server для забезпечення доступу Linux-клієнтів;
- механізми POSIX-прав та POSIX ACL для розмежування доступу;
- віртуалізоване середовище Oracle VirtualBox;
- клієнтські операційні системи Windows та Ubuntu Desktop;
- інструменти тестування iperf3, dd, robocopy, nmap, smbclient, smbstatus.

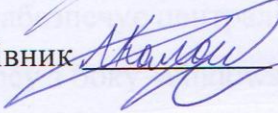
Обладнання та програмне забезпечення:

1. Віртуалізоване середовище Oracle VirtualBox для створення лабораторного стенду.
2. Віртуальна машина FileServer на базі Ubuntu Server для розгортання служб Samba та NFS.
3. Віртуальна машина UbuntuClient для перевірки доступу до NFS-ресурсу.
4. Віртуальна машина Windows VM для перевірки доступу до SMB-ресурсу.
5. Програмні компоненти: Samba, nfs-kernel-server, OpenSSH Server, acl, iperf3, nmap.
6. Засоби тестування продуктивності та функціональності: iperf3, dd, robocopy, smbclient, smbstatus, journalctl.

Консультанти з випускної кваліфікаційної роботи із зазначенням розділів, що їх стосуються

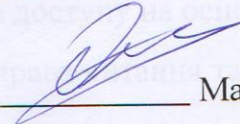
Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1, 2, 3	Миколайчук В.В.	20.02.2026 	 20.02.2026

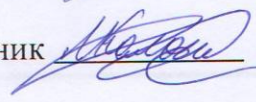
Студент  Максим ОСЕЦЬКИЙ

Керівник  Марія КОЛОЩУК

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назви етапів випускної кваліфікаційної роботи	Термін виконання	Примітка
1	Отримання та аналіз завдання, постановка мети й задач дослідження	03.02.2026 – 14.02.2026	Виконано
2	Аналіз призначення файлових серверів у гетерогенних мережах, огляд протоколів SMB/CIFS та NFS	15.02.2026 – 01.03.2026	Виконано
3	Дослідження проблем змішаного доступу, моделей прав POSIX ACL і Windows ACL	02.03.2026 – 15.03.2026	Виконано
4	Проектування архітектури файлового сервера та моделі прав доступу	16.03.2026 – 31.03.2026	Виконано
5	Розгортання серверної ОС, налаштування мережі, Samba та NFS	01.04.2026 – 20.04.2026	Виконано
6	Створення користувачів, груп, налаштування ACL та перевірка доступу з Windows і Linux	21.04.2026 – 05.05.2026	Виконано
7	Проведення функціонального, навантажувального та безпекового тестування	06.05.2026 – 20.05.2026	Виконано
8	Аналіз результатів, оформлення пояснювальної записки та висновків	21.05.2026 – 02.06.2026	Виконано
9	Захист	23.06.2026	

Студент  Максим ОСЕЦЬКИЙ

Керівник  Марія КОЛОЩУК

АНОТАЦІЯ

Пояснювальна записка до випускної кваліфікаційної роботи бакалавра містить 107 сторінок, 45 рисунків, 11 таблиць. Список використаних джерел складається з 24 найменування.

Метою кваліфікаційної роботи є проєктування файлового сервера змішаного доступу SMB/NFS на базі Linux, який забезпечує централізоване зберігання даних та доступ до спільного файлового ресурсу з боку Windows- і Linux-клієнтів.

У процесі виконання кваліфікаційної роботи отримано такі основні результати:

1. проаналізовано призначення файлових серверів у гетерогенних мережах та особливості використання протоколів SMB/CIFS і NFS;
2. розглянуто проблеми одночасного змішаного доступу до спільного каталогу, зокрема питання блокування файлів, кешування та узгодженості даних;
3. досліджено відмінності між моделями прав доступу POSIX ACL і Windows ACL, а також принципи відповідності користувачів у середовищах Linux і Windows;
4. спроектовано архітектуру лабораторного стенду файлового сервера на базі Linux із використанням віртуальних машин FileServer, UbuntuClient та Windows VM;
5. розгорнуто файловий сервер на базі Ubuntu Server, налаштовано служби Samba, NFS та OpenSSH;
6. створено спільний каталог /srv/storage/mixed, доступний для Windows-клієнта через SMB-ресурс \\192.168.56.10\\mixed і для Linux-клієнта через NFS-монтування /mnt/mixed;
7. реалізовано модель розмежування прав доступу на основі користувачів, груп і POSIX ACL, де користувач ivan має права читання та запису, а користувач petro має лише права читання;

KKIKB.KP.B. – 123 – 11 - ПЗ							
Змн.	Арк.	№ докум.	Підпис	Дата	Літ.	Арк.	Аркуші
Розроб.		Осецький М.П.		22.06.16	Проект файлового сервера змішаного доступу SMB/NFS на базі Linux		
Керівник		Колощук М.С.		22.06.16			
Реценз.		Головня О.С.		22.06.16			
Н. контр.		Колощук М.С.		22.06.16			
Затверд.		Єфіменко А.А.		22.06.16			
						4	107
					ФІКТ, Гр. КІ-22-1		

8. проведено функціональне тестування доступу з Windows- та Linux-клієнтів, перевірено змішаний доступ SMB/NFS, права користувачів і роботу механізмів блокування;
9. виконано навантажувальне тестування продуктивності мережі та файлового доступу за допомогою iperf3, dd і robocopy;
10. проведено базове тестування безпеки, зокрема перевірку відкритих портів, журналювання невдалих SSH-спроб, обмеження прав користувача retro та роботу механізму root_squash для NFS.

Ключові слова: ФАЙЛОВИЙ СЕРВЕР, LINUX, SMB, CIFS, SAMBA, NFS, POSIX ACL, ГЕТЕРОГЕННА МЕРЕЖА, ЗМІШАНИЙ ДОСТУП, WINDOWS, UBUNTU SERVER.

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

ANNOTATION

The explanatory note to the Bachelor's qualification thesis contains 107 pages, 45 figures and 11 tables. The list of references consists of 24 sources.

The aim of the qualification thesis is to design a mixed-access SMB/NFS file server based on Linux, which provides centralized data storage and access to a shared file resource from both Windows and Linux clients.

The following main results were obtained during the qualification thesis:

1. the purpose of file servers in heterogeneous networks and the features of SMB/CIFS and NFS protocols were analysed;
2. the problems of simultaneous mixed access to a shared directory were considered, including file locking, caching and data consistency issues;
3. the differences between POSIX ACL and Windows ACL access control models were examined, as well as the principles of user mapping in Linux and Windows environments;
4. the architecture of a laboratory test bench based on virtual machines FileServer, UbuntuClient and Windows VM was designed;
5. a file server based on Ubuntu Server was deployed, and Samba, NFS and OpenSSH services were configured;
6. a shared directory /srv/storage/mixed was created and made available to the Windows client through the SMB share \\192.168.56.10\\mixed and to the Linux client through the NFS mount point /mnt/mixed;
7. an access control model based on users, groups and POSIX ACL was implemented, where the user ivan has read and write permissions, while the user petro has read-only access;
8. functional testing of access from Windows and Linux clients was performed, including verification of mixed SMB/NFS access, user permissions and file locking mechanisms;
9. performance testing of the network and file access was carried out using iperf3, dd and robocopy tools;

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

10.basic security testing was performed, including open port scanning, logging of failed SSH login attempts, verification of petro user restrictions and validation of the NFS root_squash mechanism.

Keywords: FILE SERVER, LINUX, SMB, CIFS, SAMBA, NFS, POSIX ACL, HETEROGENEOUS NETWORK, MIXED ACCESS, WINDOWS, UBUNTU SERVER.

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

ВСТУП	10
РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ ПОБУДОВИ ФАЙЛОВИХ СЕРВЕРІВ У ГЕТЕРОГЕННИХ МЕРЕЖАХ	13
1.1 Призначення файлових серверів у гетерогенних мережах; протоколи SMB/CIFS (Samba) та NFS.....	13
1.2 Проблематика одночасного змішаного доступу: конфлікти блокувань файлів і узгодженість кешування	18
1.3 Розбіжності моделей прав доступу (POSIX ACL та Windows ACL) і мапінг користувачів (UID/GID ↔ SID)	22
1.4 Огляд та порівняння аналогів (TrueNAS, готові NAS-рішення, чисті Samba/NFS-конфігурації)	27
РОЗДІЛ 2 МОДЕЛЮВАННЯ ФАЙЛОВОГО СЕРВЕРА ЗМІШАНОГО ДОСТУПУ.....	33
2.1 Обґрунтування вибору дистрибутива Linux і файлової системи (ext4/XFS/ZFS); організація дискового простору.....	33
2.2 Проектування архітектури системи (структурна схема компонентів: ОС, дискова підсистема, Samba, NFS-сервер, служба автентифікації)	37
2.3 Модель прав доступу та автентифікації: інтеграція з LDAP/Active Directory/FreeIPA, узгодження ACL між протоколами	41
2.4 Сценарії використання та діаграми взаємодії клієнтів із сервером.....	47
2.5 Методика тестування системи.....	51
РОЗДІЛ 3 РОЗРОБКА ТА ТЕСТУВАННЯ СИСТЕМИ	58
3.1 Розгортання серверної ОС, підготовка дискової підсистеми та налаштування служби Samba	58
3.2 Налаштування NFS-сервера та механізмів спільного змішаного доступу і блокувань.....	63
3.3 Функціональне тестування (доступ із Windows- та Linux-клієнтів, перевірка прав, одночасний запис)	68
3.4 Навантажувальне тестування продуктивності (методика, інструменти, швидкість читання/запису під навантаженням).....	76
3.5 Тестування безпеки та відмовостійкості	82
ВИСНОВКИ.....	93
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	97
ДОДАТКИ.....	99

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				8
Змн.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ACL — Access Control List, список контролю доступу

AD — Active Directory, служба каталогів Microsoft

CIFS — Common Internet File System, мережева файлова система Microsoft

GID — Group Identifier, ідентифікатор групи в Unix/Linux

IP — Internet Protocol, міжмережевий протокол

LDAP — Lightweight Directory Access Protocol, протокол доступу до служб

NAS — Network Attached Storage, мережеве файлове сховище

NFS — Network File System, мережева файлова система Unix/Linux

ОС — операційна система

ПЗ — програмне забезпечення

POSIX — Portable Operating System Interface, стандарт сумісності Unix-подібних систем

SID — Security Identifier, ідентифікатор безпеки Windows

SMB — Server Message Block, протокол спільного доступу до файлів у мережі

SSH — Secure Shell, захищений протокол віддаленого адміністрування

TCP — Transmission Control Protocol, протокол керування передаванням

UID — User Identifier, ідентифікатор користувача в Unix/Linux

VM — Virtual Machine, віртуальна машина

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		9

ВСТУП

У сучасних інформаційних системах файлові сервери залишаються одним із ключових елементів мережевої інфраструктури. Вони забезпечують централізоване зберігання даних, організацію спільної роботи користувачів, контроль доступу до файлів, резервування інформації та спрощення адміністрування. Особливо важливим є використання файлових серверів у гетерогенних мережах, де одночасно працюють клієнтські пристрої з різними операційними системами, зокрема Windows та Linux.

У Windows-середовищі для організації спільного доступу до файлів традиційно використовується протокол SMB/CIFS. У Linux цей протокол реалізується за допомогою програмного комплексу Samba, який дозволяє створювати спільні мережеві ресурси, автентифікувати користувачів і надавати доступ до файлів із Windows-клієнтів. Для Linux- та Unix-подібних систем поширеним рішенням є протокол NFS, який дає змогу монтувати віддалені каталоги у локальну файлову систему клієнта та працювати з ними як зі звичайними каталогами.

Актуальність теми зумовлена потребою у створенні універсального файлового сервера, який може одночасно обслуговувати Windows- та Linux-клієнтів, забезпечуючи єдине централізоване сховище даних. Такий підхід є корисним для навчальних лабораторій, невеликих офісів, тестових середовищ, організацій із комбінованою інфраструктурою, а також для випадків, коли необхідно забезпечити сумісність між різними операційними системами без використання дорогих комерційних NAS-рішень.

Разом із тим реалізація змішаного доступу SMB/NFS має низку технічних особливостей. Насамперед необхідно узгодити моделі прав доступу Windows і Linux, оскільки Windows використовує власну модель безпеки на основі SID та ACL, тоді як Linux базується на UID, GID, класичних POSIX-правах і POSIX ACL. Також важливими є питання блокування файлів, кешування, одночасного доступу до одного каталогу різними клієнтами, захисту від несанкціонованого доступу та забезпечення стабільної роботи після перезавантаження серверної і клієнтських

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

систем.

Метою кваліфікаційної роботи є проєктування файлового сервера змішаного доступу SMB/NFS на базі Linux, який забезпечує централізований доступ до спільного файлового каталогу для Windows- та Linux-клієнтів із реалізацією розмежування прав доступу, функціонального тестування, оцінювання продуктивності та перевірки базових механізмів безпеки.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- проаналізувати призначення файлових серверів у гетерогенних мережах;
- розглянути особливості протоколів SMB/CIFS та NFS;
- дослідити проблеми одночасного змішаного доступу, блокування файлів і кешування;
- порівняти моделі прав доступу POSIX ACL та Windows ACL;
- розглянути можливості мапінгу користувачів між Linux- і Windows-середовищами;
- виконати огляд аналогів, зокрема TrueNAS, готових NAS-рішень і чистих Samba/NFS-конфігурацій;
- обґрунтувати вибір операційної системи Linux, файлової системи та програмних компонентів;
- спроектувати архітектуру файлового сервера змішаного доступу;
- розгорнути серверну операційну систему та налаштувати мережеву взаємодію;
- налаштувати Samba для доступу Windows-клієнтів;
- налаштувати NFS для доступу Linux-клієнтів;
- створити користувачів, групи та POSIX ACL для розмежування доступу;
- провести функціональне тестування доступу з Windows- та Linux-клієнтів;
- виконати навантажувальне тестування продуктивності;
- провести базове тестування безпеки та відмовостійкості системи.

Об'єктом дослідження є процес організації спільного файлового доступу в гетерогенній комп'ютерній мережі.

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		11

Предметом дослідження є методи, протоколи та програмні засоби побудови файлового сервера змішаного доступу SMB/NFS на базі Linux.

У межах кваліфікаційної роботи було розгорнуто лабораторний стенд у середовищі Oracle VirtualBox. До складу стенду увійшли три основні віртуальні машини: FileServer на базі Ubuntu Server, UbuntuClient на базі Ubuntu Desktop та Windows VM на базі Windows. Для обміну між вузлами використано ізольовану Host-only мережу 192.168.56.0/24. Файловий сервер отримав IP-адресу 192.168.56.10, Linux-клієнт — 192.168.56.20, а Windows-клієнт — 192.168.56.30.

Практичним результатом роботи є працездатний файловий сервер зі спільним каталогом /srv/storage/mixed. Для Windows-клієнта каталог доступний через SMB-ресурс \\192.168.56.10\\mixed, а для Linux-клієнта — через NFS-монтування /mnt/mixed. У системі реалізовано розмежування прав доступу: користувач ivan має права читання і запису, користувач retro має лише право читання, а адміністратор має розширені права керування ресурсом.

Під час тестування було підтверджено працездатність доступу через SMB і NFS, перевірено створення файлів із різних клієнтів, видимість змін у спільному каталозі, обмеження доступу для користувача з правами лише читання, роботу механізмів блокування SMB, продуктивність мережі та файлового доступу, а також базові аспекти безпеки, зокрема журналювання невдалих SSH-спроб і перевірку відкритих портів сервера.

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		12

РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ ПОБУДОВИ ФАЙЛОВИХ СЕРВЕРІВ У ГЕТЕРОГЕННИХ МЕРЕЖАХ

1.1 Призначення файлових серверів у гетерогенних мережах; протоколи SMB/CIFS (Samba) та NFS

Файловий сервер є одним із базових компонентів мережевої інфраструктури, призначеним для централізованого зберігання, обробки та надання доступу до файлів користувачам і прикладним системам. Його основна функція полягає в тому, щоб забезпечити спільне використання даних у межах локальної або корпоративної мережі. Завдяки файловому серверу користувачі можуть працювати з документами, каталогами, архівами, резервними копіями та іншими файлами незалежно від того, на якому фізичному носії вони зберігаються.

У невеликих мережах файли часто зберігаються безпосередньо на робочих станціях користувачів. Такий підхід є простим, однак він має значні недоліки. По-перше, ускладнюється централізоване резервне копіювання, оскільки дані розміщені на різних комп'ютерах. По-друге, адміністратору складніше контролювати права доступу, адже кожна робоча станція може мати власні локальні налаштування. По-третє, доступність файлів залежить від стану конкретного комп'ютера: якщо він вимкнений або несправний, інші користувачі втрачають доступ до потрібних даних. Файловий сервер усуває ці проблеми, оскільки всі важливі файли зберігаються централізовано, а керування доступом виконується з одного місця.

Використання файлового сервера також підвищує рівень безпеки. Адміністратор може створювати користувачів і групи, призначати права читання, запису або адміністрування, обмежувати доступ до окремих каталогів, вести журнали подій та контролювати активність користувачів. У разі потреби можна організувати резервне копіювання, відновлення даних після збоїв, шифрування або підключення додаткових засобів моніторингу. Саме тому файлові сервери широко використовуються в офісних, навчальних, виробничих і корпоративних середовищах.

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		13

Особливо важливим є використання файлових серверів у гетерогенних мережах. Гетерогенною називається мережа, у якій одночасно використовуються різні операційні системи, протоколи, апаратні платформи та програмні рішення. Типовим прикладом є інфраструктура, де частина користувачів працює на операційній системі Windows, а інша частина — на Linux. У такому середовищі виникає потреба в єдиному файловому ресурсі, який буде доступний для обох типів клієнтів.

У межах теми кваліфікаційної роботи гетерогенна мережа розглядається як середовище, у якому один файловий сервер на базі Linux обслуговує Windows- та Linux-клієнтів. Для Windows-клієнтів використовується протокол SMB/CIFS, а для Linux-клієнтів — протокол NFS. Обидва протоколи надають доступ до одного спільного серверного каталогу, однак використовують різні механізми автентифікації, передачі даних, кешування та контролю прав доступу.



Рисунок 1.1 — Узагальнена схема файлового сервера змішаного доступу в гетерогенній мережі

Одним із найпоширеніших протоколів доступу до мережеских файлів у Windows-середовищі є SMB, або Server Message Block. Цей протокол використовується для організації доступу до спільних папок, файлів, принтерів та інших мережеских ресурсів. Для користувача Windows такий ресурс зазвичай виглядає як мережева папка або підключений мережеский диск. Наприклад, шлях

\\192.168.56.10\\mixed означає доступ до спільного ресурсу mixed на сервері з IP-адресою 192.168.56.10.

CIFS, або Common Internet File System, історично є розвитком ранніх версій SMB. У сучасній практиці терміни SMB і CIFS часто використовуються разом, хоча більш коректним є використання назви SMB для сучасних реалізацій протоколу. Протокол SMB підтримує автентифікацію користувачів, блокування файлів, керування мережевими ресурсами, передачу метаданих і роботу з правами доступу. Завдяки цьому він став стандартним рішенням для файлового обміну в Windows-мережах.

У Linux-середовищі підтримка SMB/CIFS реалізується за допомогою програмного комплексу Samba. Samba дозволяє Linux-серверу виступати як файловий сервер для Windows-клієнтів. За допомогою Samba можна створювати спільні каталоги, налаштовувати автентифікацію, задавати права доступу, обмежувати запис або читання, а також інтегрувати сервер у Windows-орієнтовану інфраструктуру. У простішому варіанті Samba може використовувати локальні облікові записи, а в складніших середовищах — взаємодіяти з доменними службами.

Перевагою Samba є її універсальність. Вона дозволяє використовувати Linux як повноцінний файловий сервер для Windows-клієнтів без потреби встановлювати Windows Server. Це особливо важливо у випадках, коли необхідно побудувати економічне, гнучке і контрольоване рішення на базі відкритого програмного забезпечення. Крім того, Samba добре поєднується з Linux-файловими системами та механізмами POSIX ACL, що дозволяє реалізувати досить гнучку модель прав доступу.

Іншим важливим протоколом файлового доступу є NFS, або Network File System. Він традиційно використовується в Unix- та Linux-середовищах. Головна ідея NFS полягає в тому, що віддалений каталог сервера монтується в локальну файлову систему клієнта. Після цього користувач може працювати з віддаленим каталогом майже так само, як зі звичайним локальним каталогом. Наприклад,

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		15

серверний каталог `/srv/storage/mixed` може бути підключений на клієнтській машині як `/mnt/mixed`.

NFS є зручним рішенням для Linux-клієнтів, оскільки природно інтегрується з POSIX-моделлю прав доступу. Для визначення власника файлів і груп використовуються UID та GID. Це означає, що сервер і клієнт мають однаково розуміти, який користувач або група відповідає певному числовому ідентифікатору. Саме тому під час побудови NFS-середовища важливо узгоджувати користувачів і групи між сервером та клієнтом або використовувати централізовану систему автентифікації.

Перевагою NFS є простота інтеграції з Linux-системами, висока швидкість роботи в Unix-подібному середовищі та зручність автоматичного монтування ресурсів. Недоліком є те, що NFS менш природний для Windows-клієнтів і потребує уважного налаштування прав доступу. Крім того, під час використання NFS необхідно враховувати особливості кешування, блокування файлів і безпеки, зокрема механізм `root_squash`, який обмежує права `root`-користувача клієнтської машини на серверному ресурсі.

Побудова файлового сервера змішаного доступу передбачає одночасне використання SMB/CIFS і NFS для одного спільного каталогу. У такій архітектурі Windows-клієнти працюють із сервером через Samba, а Linux-клієнти — через NFS. На сервері фактично існує один фізичний каталог, наприклад `/srv/storage/mixed`, але для різних клієнтів він надається через різні протоколи. Це дозволяє створити спільний файловий простір для користувачів різних операційних систем.

Разом із тим такий підхід потребує уважного проектування. Необхідно забезпечити узгодженість прав доступу, правильне створення користувачів і груп, налаштування ACL, коректне успадкування прав для нових файлів, а також перевірити поведінку системи під час одночасного доступу. Якщо ці аспекти не врахувати, може виникнути ситуація, коли файл, створений через SMB, буде недоступний через NFS, або навпаки.

У межах цієї кваліфікаційної роботи файловий сервер змішаного доступу реалізується на базі Linux. Для доступу Windows-клієнта використовується Samba,

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		16

а для доступу Linux-клієнта — NFS. Як спільний ресурс використовується каталог /srv/storage/mixed. Така архітектура дозволяє дослідити принципи побудови файлового сервера SMB/NFS, перевірити доступ із різних операційних систем і оцінити працездатність запропонованого рішення.

Таблиця 1.1

Порівняння протоколів SMB/CIFS та NFS

Критерій порівняння	SMB/CIFS	NFS
Основне середовище використання	Windows-мережі	Linux-мережі
Реалізація на Linux	Samba	nfs-kernel-server
Спосіб доступу	Мережева папка або мережевий диск	Монтування в локальну файлову систему
Приклад доступу	\\192.168.56.10\\mixed	/mnt/mixed
Типова серверна директорія	Каталог, опублікований у smb.conf	Каталог, експортований через /etc/exports
Модель ідентифікації	Користувачі Windows/Samba, SID	UID/GID
Типові клієнти	Windows, Linux із SMB-клієнтом	Linux
Переваги	Зручність для Windows, підтримка мережних дисків, автентифікації та блокувань	Природна інтеграція з Linux, висока швидкість, просте монтування
Недоліки	Потребує узгодження з POSIX-правами на Linux-сервері	Потребує узгодження UID/GID і врахування кешування
Доцільність використання	Надання доступу Windows-клієнтам до спільних ресурсів	Надання доступу Linux-клієнтам до серверних каталогів

Отже, SMB/CIFS і NFS мають спільне призначення — забезпечення мережевого доступу до файлів, однак орієнтовані на різні клієнтські середовища. Для побудови файлового сервера змішаного доступу доцільним є поєднання обох протоколів на одному Linux-сервері. Це дозволяє створити єдиний спільний файловий ресурс для Windows- та Linux-клієнтів, зберігаючи при цьому переваги кожного протоколу.

1.2 Проблематика одночасного змішаного доступу: конфлікти блокувань файлів і узгодженість кешування

Під час побудови файлового сервера змішаного доступу недостатньо лише налаштувати окремо Samba та NFS. Основна складність полягає в тому, що обидва протоколи можуть надавати доступ до одного й того самого каталогу, однак використовують різні механізми роботи з файлами. Якщо Windows-клієнт відкриває файл через SMB, а Linux-клієнт у той самий час працює з цим самим файлом через NFS, виникає ситуація одночасного змішаного доступу. Така ситуація потребує окремого аналізу, оскільки різні протоколи можуть по-різному обробляти блокування, кешування, оновлення метаданих і момент фіксації змін.

У звичайному випадку, коли всі клієнти працюють через один протокол, сервер має єдину логіку обробки доступу. Наприклад, якщо всі користувачі підключаються через SMB, Samba контролює відкриті файли, активні підключення, блокування та кешування з урахуванням поведінки Windows-клієнтів. Якщо всі клієнти працюють через NFS, то використовується інша модель доступу, орієнтована на Unix/Linux-системи. Проблема виникає тоді, коли один каталог одночасно експортується через SMB і NFS, оскільки доступ до одних і тих самих файлів здійснюється через різні протокольні механізми.

Блокування файлів є одним із ключових механізмів, який використовується для запобігання пошкодженню або перезапису даних. Його призначення полягає в тому, щоб регулювати ситуації, коли кілька користувачів або програм одночасно відкривають один файл. Якщо один клієнт редагує файл, сервер або клієнтська програма можуть обмежити запис з боку інших користувачів, попередити про конфлікт або дозволити лише читання. У Windows-середовищі такі механізми є особливо важливими для роботи офісних документів, баз даних, проєктних файлів та інших ресурсів, які можуть одночасно відкриватися кількома користувачами.

У протоколі SMB підтримуються механізми блокування та оптимістичного кешування, зокрема `oplocks` і `leases`. Вони дозволяють клієнту тимчасово кешувати дані локально та зменшувати кількість звернень до сервера. Якщо інший клієнт намагається отримати доступ до того самого файлу, сервер може повідомити

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		18

першого клієнта про необхідність синхронізувати або скинути кешовані зміни. Такий підхід підвищує продуктивність, але потребує коректної взаємодії між клієнтами та сервером.

У NFS поведінка відрізняється. NFS орієнтований на модель, у якій віддалений каталог монтується в локальну файлову систему клієнта. Linux-клієнт після монтування працює з каталогом так, ніби він є частиною локальної файлової ієрархії. Для підвищення продуктивності NFS-клієнт також може кешувати дані та атрибути файлів. У багатьох випадках використовується принцип close-to-open consistency, за якого зміни, внесені одним клієнтом, гарантовано стають видимими іншому клієнту після закриття файлу та його повторного відкриття. Однак миттєва синхронізація в усіх сценаріях не завжди гарантується.

Проблема узгодженості кешування полягає в тому, що клієнт може певний час працювати з локально збереженими даними або метаданими, не звертаючись до сервера при кожній операції. Це підвищує швидкість роботи, але може створювати затримку у відображенні змін. Наприклад, файл був змінений через SMB-клієнт, але NFS-клієнт ще певний час може бачити старі атрибути або старий вміст, якщо його кеш не оновився. У більшості звичайних сценаріїв така затримка є короткою, однак для систем із високими вимогами до узгодженості даних це може бути критичним.

Окрему складність створює одночасне редагування одного файлу через різні протоколи. Наприклад, користувач Windows відкриває файл через ресурс \\192.168.56.10\\mixed, а користувач Linux у цей самий час відкриває його через каталог /mnt/mixed. Якщо обидва користувачі вносять зміни та зберігають файл незалежно один від одного, може виникнути конфлікт. У найпростішому випадку останнє збереження перезапише попередні зміни. У складніших випадках файл може опинитися в непередбачуваному стані, особливо якщо з ним працюють не текстові редактори, а прикладні програми, які використовують власний формат збереження.

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				19
Змн.	Арк.	№ докум.	Підпис	Дата		

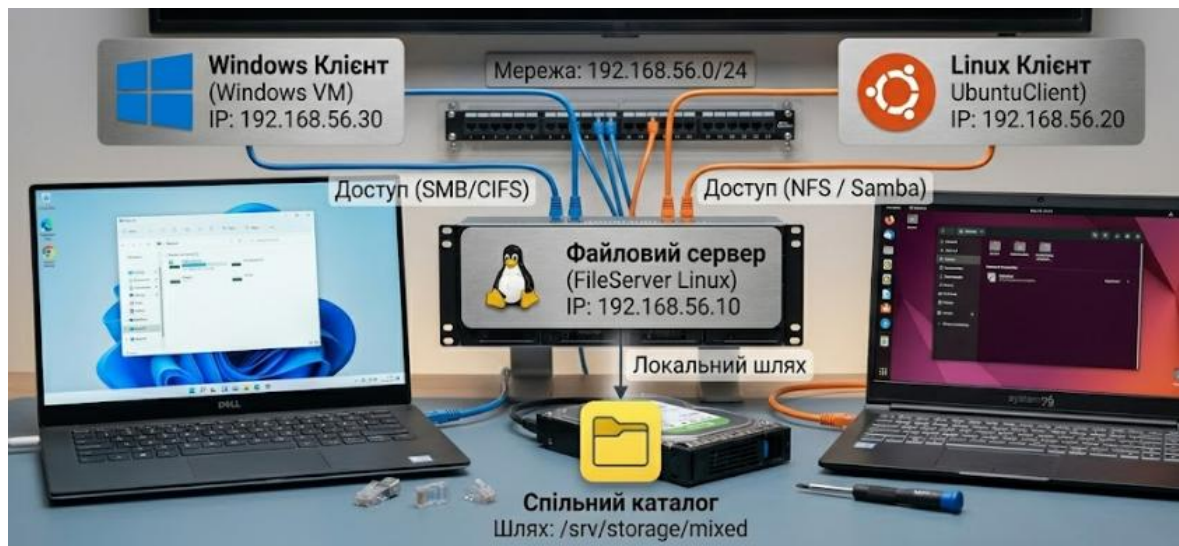


Рисунок 1.2 — Сценарій одночасного доступу SMB- та NFS-клієнтів до одного файлу

Для зменшення ризику конфліктів під час змішаного доступу необхідно застосовувати кілька підходів. Перший підхід полягає в чіткому розмежуванні прав доступу. Користувачі, які не повинні змінювати файли, мають отримувати лише права читання. У практичній реалізації цієї роботи користувач `ivan` має права читання і запису, а користувач `retro` має тільки права читання. Така модель дозволяє зменшити ймовірність випадкової зміни або видалення даних користувачем, який не повинен виконувати такі операції.

Другий підхід полягає в правильному налаштуванні прав на рівні Linux-файлової системи. Оскільки і Samba, і NFS працюють з одним серверним каталогом, фактичні права доступу визначаються Linux-правами та POSIX ACL. Якщо права на каталог `/srv/storage/mixed` налаштовані некоректно, можуть виникнути ситуації, коли файл, створений через SMB, буде недоступний через NFS, або файл, створений через NFS, не зможе бути змінений через SMB. Тому важливо налаштувати не лише звичайні права власника й групи, а й ACL та default ACL для автоматичного успадкування прав новими файлами.

Третій підхід полягає в організаційному обмеженні одночасного редагування критичних файлів. Навіть за наявності технічних механізмів блокування адміністратор повинен враховувати, що SMB і NFS не завжди однаково інтерпретують стан відкритого файлу. Тому для важливих документів, баз даних

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				20
Змн.	Арк.	№ докум.	Підпис	Дата		

або конфігураційних файлів бажано визначати правила роботи: не редагувати один і той самий файл одночасно через різні протоколи, використовувати окремі каталоги для різних груп користувачів або застосовувати прикладні системи, які самостійно забезпечують контроль версій і блокування.

У Samba для аналізу активних підключень і блокувань використовується команда `smbstatus`. Вона дозволяє переглянути список активних SMB-сесій, відкритих файлів, ідентифікатори процесів, користувачів, режими доступу та наявність механізмів `lease/oplock`. У практичній частині роботи команда `smbstatus --locks` була використана для перевірки того, що Samba фіксує відкриті файли та застосовує механізми контролю доступу з боку SMB-клієнта. Отриманий результат показав наявність активного SMB `lease` для відкритого файлу, що підтверджує роботу механізму відстеження відкритих файлів.

Водночас потрібно враховувати, що інформація про SMB-блокування відображає саме стан доступу з боку SMB-клієнтів. NFS-клієнти можуть працювати з тим самим файлом через інший механізм доступу. Саме тому на практиці змішаний доступ SMB/NFS вимагає уважного тестування. Необхідно перевірити не лише сам факт доступу, а й поведінку системи під час створення файлів, редагування, видалення, читання з іншого клієнта та спроби запису користувачами з різними правами.

У межах цієї кваліфікаційної роботи змішаний доступ перевірявся на прикладі одного спільного каталогу `/srv/storage/mixed`. Windows-клієнт отримував доступ до нього через SMB-ресурс `\192.168.56.10\mixed`, а Linux-клієнт — через NFS-монтування `/mnt/mixed`. Було перевірено, що файл, створений через Windows-клієнт, відображається на Linux-клієнті, а файл, створений через Linux-клієнт, відображається у Windows. Це підтверджує, що обидва протоколи працюють з одним і тим самим серверним каталогом.

Таким чином, проблема одночасного змішаного доступу є однією з центральних під час проектування файлового сервера SMB/NFS. Вона включає питання блокування файлів, кешування, узгодженості змін, правильного налаштування ACL і обмеження прав користувачів. Для лабораторного стенду ці

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				21
Змн.	Арк.	№ докум.	Підпис	Дата		

аспекти дозволяють продемонструвати принцип роботи змішаного доступу та визначити обмеження такого підходу. Для реального корпоративного використання додатково необхідно враховувати навантаження, кількість одночасних клієнтів, типи файлів, політики резервного копіювання та правила одночасного редагування.

1.3 Розбіжності моделей прав доступу (POSIX ACL та Windows ACL) і мапінг користувачів (UID/GID ↔ SID)

Одним із ключових питань під час проектування файлового сервера змішаного доступу SMB/NFS є узгодження моделей прав доступу. Windows- та Linux-системи використовують різні підходи до ідентифікації користувачів, груп і дозволів на файли. Через це адміністратор повинен не лише налаштувати мережеві протоколи, а й забезпечити правильне зіставлення користувачів і прав доступу на рівні серверної файлової системи.

У Linux використовується модель доступу, заснована на користувачах, групах і правах POSIX. Кожен файл або каталог має власника, групу-власника та набір дозволів для трьох категорій: власника, групи та інших користувачів. Базові дозволи позначаються як *r*, *w* і *x*, тобто читання, запис і виконання. Для каталогів ці права мають дещо інше практичне значення: право читання дозволяє переглядати список файлів, право запису — створювати або видаляти об'єкти, а право виконання — переходити до каталогу та звертатися до його вмісту.

Класична POSIX-модель є простою та зрозумілою, однак у складніших системах її можливостей часто недостатньо. Наприклад, може виникнути потреба надати одній групі користувачів право читання і запису, іншій групі — лише читання, а адміністраторам — повний доступ. Для цього використовуються POSIX ACL, або списки контролю доступу. POSIX ACL дозволяють призначати права не лише власнику, групі та іншим користувачам, а й конкретним користувачам або додатковим групам.

У Windows модель доступу побудована інакше. Користувачі та групи ідентифікуються не через UID і GID, а через SID — Security Identifier. SID є

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				22
Змн.	Арк.	№ докум.	Підпис	Дата		

унікальним ідентифікатором безпеки, за яким система визначає конкретного користувача або групу. Права доступу до файлів і каталогів у Windows задаються за допомогою Windows ACL. Така модель дозволяє дуже детально визначати дозволи: читання, запис, зміну, виконання, повний доступ, успадкування прав та інші параметри.

Основна відмінність між POSIX ACL та Windows ACL полягає в тому, що Linux-файлова система природно працює з UID/GID та POSIX-правами, а Windows-клієнти очікують модель доступу, наближену до Windows ACL. Коли Windows-клієнт підключається до Linux-сервера через Samba, він працює з SMB-ресурсом як зі звичайною мережевою папкою Windows. Однак фактично файли зберігаються на Linux-файловій системі, тому остаточне рішення щодо доступу приймається з урахуванням Linux-користувачів, груп і POSIX ACL.

Samba виконує роль проміжного рівня між Windows-клієнтом і Linux-файловою системою. Коли користувач Windows підключається до SMB-ресурсу, Samba автентифікує його та зіставляє з відповідним локальним користувачем Linux. У простій локальній моделі для цього створюються однакові облікові записи в Linux і Samba. Наприклад, якщо у системі є Linux-користувач ivan, то для доступу через SMB також створюється Samba-користувач ivan. Після цього користувач, який підключається з Windows як ivan, на рівні Linux-файлової системи працює від імені відповідного Linux-користувача.

У випадку NFS ситуація має іншу особливість. NFS орієнтується на числові ідентифікатори UID і GID. Це означає, що сервер і клієнт повинні однаково розуміти, який користувач відповідає певному UID, а яка група — певному GID. Якщо на сервері користувач ivan має UID 1101, а на клієнті UID 1101 належить іншому користувачу, можуть виникнути помилки доступу або некоректне визначення власника файлів. Тому для NFS важливим є узгодження UID/GID між сервером і клієнтами.

У великих організаціях проблема зіставлення користувачів вирішується за допомогою централізованих служб каталогів. Для цього можуть використовуватися LDAP, Active Directory або FreeIPA. Такі системи дозволяють

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				23
Змн.	Арк.	№ докум.	Підпис	Дата		

зберігати облікові записи користувачів і груп централізовано, а сервери та клієнти отримують єдине джерело ідентифікації. У результаті адміністратору не потрібно вручну створювати однакових користувачів на кожній машині, а ризик невідповідності UID/GID або облікових записів зменшується.

У межах цієї кваліфікаційної роботи використано локальну модель автентифікації. Це означає, що користувачі та групи створювалися безпосередньо на файловому сервері та Linux-клієнті. Такий підхід є доцільним для лабораторного стенду, оскільки він дозволяє наочно продемонструвати принципи роботи SMB, NFS, POSIX ACL і мапінгу користувачів без додаткової складності, пов'язаної з розгортанням Active Directory, LDAP або FreeIPA. Інтеграція з цими службами розглядається як можливий напрям масштабування системи для корпоративного середовища.

У практичній реалізації було створено користувачів `ivan` і `petro`. Користувач `ivan` виконує роль користувача з правами читання та запису, а користувач `petro` — роль користувача з правами лише читання. Для групового керування доступом було створено групи `fs_data`, `fs_rw`, `fs_ro` та `fs_admins`. Група `fs_rw` призначена для користувачів, які мають право запису до спільного каталогу, група `fs_ro` — для користувачів із правом лише читання, а `fs_admins` — для адміністраторів файлового ресурсу.

Спільний каталог `/srv/storage/mixed` є центральною точкою, до якої застосовуються права доступу. Саме цей каталог публікується через Samba для Windows-клієнта та експортується через NFS для Linux-клієнта. Тому права на цей каталог повинні бути налаштовані так, щоб обидва протоколи працювали узгоджено. Для цього було використано POSIX ACL, які дозволяють одночасно надати різним групам різні права доступу.

Важливим елементом налаштування є default ACL. Звичайні ACL визначають права для вже існуючого каталогу або файлу, а default ACL визначають, які права будуть автоматично успадковувати нові файли та підкаталоги. Це особливо важливо для змішаного доступу. Наприклад, якщо користувач створює файл через SMB, цей файл повинен залишатися доступним для користувачів, які працюють

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		24

через NFS. Без правильного успадкування прав можуть виникнути ситуації, коли новий файл доступний тільки його власнику, але не доступний іншим дозволеним користувачам.

Узгодження прав доступу між SMB і NFS можна подати як багаторівневий процес. На першому рівні користувач проходить автентифікацію: через Samba для SMB або через локальний UID/GID для NFS. На другому рівні користувач зіставляється з Linux-користувачем і групами. На третьому рівні Linux-файлова система перевіряє права доступу до конкретного файлу або каталогу. Саме тому коректність роботи залежить не лише від конфігурації Samba чи NFS, а й від правильно побудованої моделі користувачів і POSIX ACL.

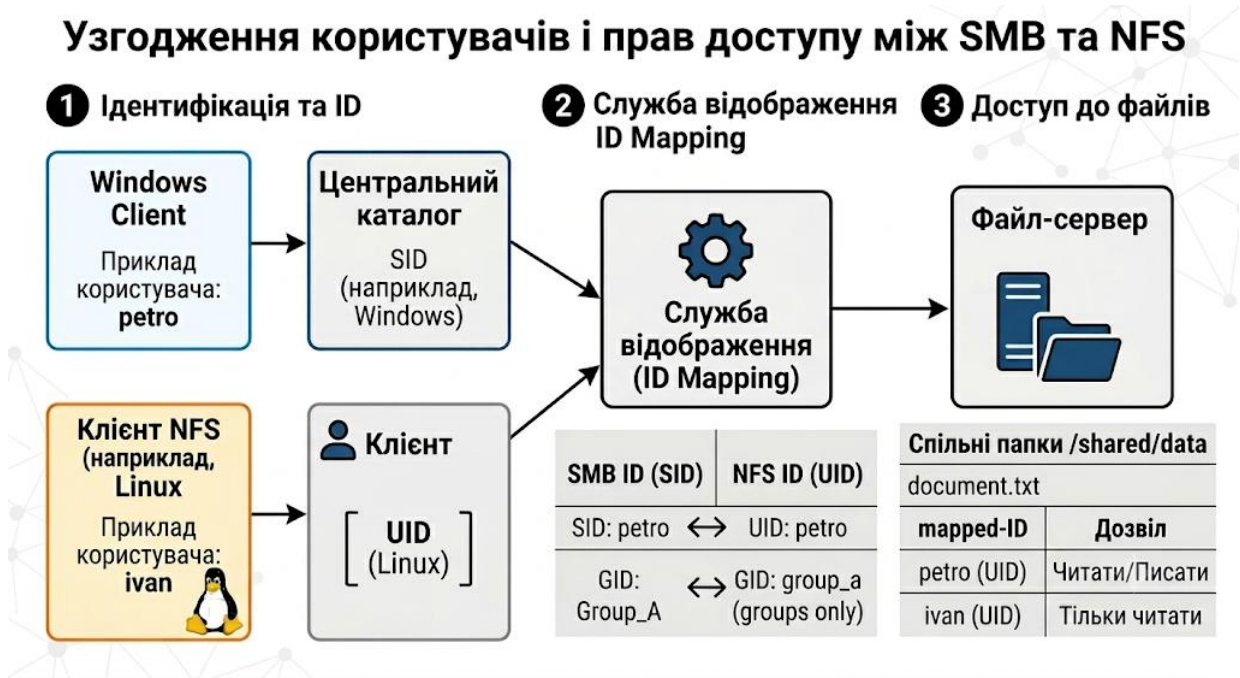


Рисунок 1.3 — Узгодження користувачів і прав доступу між SMB та NFS

Порівняння моделей POSIX ACL та Windows ACL

Критерій порівняння	POSIX ACL	Windows ACL
Основне середовище використання	Linux	Windows
Ідентифікація користувача	UID	SID
Базові права	read, write, execute	read, write, modify, full control та інші
Рівень деталізації	Середній	Високий
Механізм спадкування	default ACL для каталогів	успадкування ACL від батьківських об'єктів
Використання в NFS	Природна інтеграція	Непряме
Ідентифікація групи	GID	SID групи
Використання в SMB/Samba	Через Linux-права та Samba-мапінг	Природна модель для Windows-клієнтів
Основна складність	Необхідність узгодження UID/GID	Необхідність мапінгу на POSIX-права
Доцільність у змішаному доступі	Центральна модель прав на Linux-сервері	Зручна модель для Windows-користувачів

З таблиці видно, що POSIX ACL і Windows ACL мають спільну мету — обмеження доступу до файлів і каталогів, але реалізують її різними способами. У файлового сервера змішаного доступу центральною є Linux-модель прав, оскільки саме Linux-сервер фізично зберігає файли. Samba забезпечує доступ Windows-клієнтів до цього ресурсу, але в кінцевому результаті операції читання, запису або видалення перевіряються через права Linux-файлової системи.

Таким чином, для побудови файлового сервера SMB/NFS на базі Linux необхідно правильно визначити користувачів, групи, UID/GID, Samba-облікові записи та POSIX ACL. У межах лабораторного стенду використання локальної моделі автентифікації є достатнім, оскільки воно дозволяє реалізувати контроль доступу та перевірити роботу користувачів із різними правами. Для масштабування рішення в корпоративному середовищі доцільним є подальше використання LDAP, Active Directory або FreeIPA як централізованого джерела облікових записів.

1.4 Огляд та порівняння аналогів (TrueNAS, готові NAS-рішення, чисті Samba/NFS-конфігурації)

Під час проєктування файлового сервера змішаного доступу SMB/NFS важливо розглянути не лише ручну реалізацію на базі Linux, а й існуючі альтернативні підходи. На практиці для організації мережевого файлового сховища можуть використовуватися спеціалізовані NAS-дистрибутиви, готові апаратні NAS-пристрої або самостійно налаштовані сервери на базі Linux із використанням Samba та NFS. Кожен із цих варіантів має власні переваги, обмеження, рівень складності адміністрування та сферу доцільного застосування.

Одним із найвідоміших спеціалізованих рішень для побудови мережевого сховища є TrueNAS. Це програмна платформа, орієнтована на створення NAS-серверів із підтримкою спільного доступу до файлів, керування дисковими пулами, знімків файлової системи, користувачів, груп і мережевих сервісів. TrueNAS дозволяє надавати доступ до сховища через SMB, NFS та інші протоколи, а також має зручний веб-інтерфейс для адміністрування. Завдяки цьому адміністратор може виконувати більшість операцій без ручного редагування конфігураційних файлів.

Перевагою TrueNAS є комплексність. Система вже містить багато готових механізмів для створення мережевого сховища: керування дисками, моніторинг стану, налаштування спільних ресурсів, резервування, знімки, користувачі, групи та мережеві протоколи. Це робить TrueNAS зручним рішенням для домашніх лабораторій, невеликих офісів і середовищ, де потрібно швидко розгорнути файлове сховище з мінімальною кількістю ручних налаштувань.

Водночас TrueNAS не завжди є оптимальним варіантом для навчальної або дослідницької роботи. Через наявність готового веб-інтерфейсу частина внутрішніх механізмів прихована від користувача. Адміністратор бачить результат налаштування, але не завжди детально працює з конфігураційними файлами Samba, NFS, правами Linux і POSIX ACL. Для кваліфікаційної роботи, метою якої є саме проєктування файлового сервера SMB/NFS на базі Linux, важливо продемонструвати розуміння внутрішньої логіки роботи служб, користувачів, груп

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		27

і прав доступу. Тому використання готової NAS-платформи могло б зменшити навчальну цінність практичної частини.

Іншим варіантом є готові апаратні NAS-рішення, наприклад пристрої Synology, QNAP або аналогічні системи інших виробників. Такі рішення поєднують апаратну платформу, дискову підсистему та спеціалізоване програмне забезпечення. Вони зазвичай мають зручний графічний інтерфейс, підтримують SMB, NFS, FTP, резервне копіювання, користувачів, групи, RAID-масиви та додаткові сервіси. Для кінцевого користувача це зручний варіант, оскільки пристрій можна швидко підключити до мережі й налаштувати через веб-панель.

Готові NAS-пристрої мають кілька переваг. Вони прості в експлуатації, не потребують глибоких знань Linux-адміністрування, зазвичай мають документацію виробника та автоматичні оновлення. Такі пристрої добре підходять для невеликих організацій, де потрібно швидко отримати готове файлове сховище без складного ручного налаштування. Крім того, багато виробників пропонують додаткові функції: синхронізацію з хмарними сервісами, резервне копіювання робочих станцій, керування доступом через веб-інтерфейс та мобільні додатки.

Основним недоліком готових NAS-рішень є їхня вартість і залежність від виробника. Користувач обмежений можливостями конкретної моделі, доступними оновленнями та політикою підтримки. Крім того, частина внутрішніх механізмів може бути закритою або малодоступною для детального аналізу. Для навчального стенду це є суттєвим обмеженням, оскільки метою роботи є не лише отримання готового результату, а й дослідження принципів побудови файлового сервера змішаного доступу.

Третім підходом є ручне розгортання файлового сервера на базі Linux із використанням Samba та NFS. У такому випадку адміністратор самостійно встановлює серверну операційну систему, налаштовує мережеві інтерфейси, створює користувачів і групи, встановлює необхідні пакети, редагує конфігураційні файли Samba та NFS, налаштовує права доступу та виконує тестування. Цей підхід є більш трудомістким, але забезпечує максимальну гнучкість і повний контроль над конфігурацією.

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		28

Перевагою чистої Samba/NFS-конфігурації є те, що адміністратор безпосередньо працює з усіма ключовими компонентами системи. Наприклад, спільний SMB-ресурс описується у файлі `/etc/samba/smb.conf`, NFS-експорт — у файлі `/etc/exports`, автоматичне монтування на Linux-клієнті — у `/etc/fstab`, а права доступу перевіряються через стандартні Linux-команди `chmod`, `chown`, `setfacl` і `getfacl`. Такий підхід дозволяє точно розуміти, як саме працює кожен рівень системи.

Для кваліфікаційної роботи ручна реалізація має значну перевагу, оскільки дозволяє продемонструвати всі етапи побудови файлового сервера: від проектування архітектури до перевірки доступу з Windows- та Linux-клієнтів. Крім того, саме цей підхід дає можливість показати роботу POSIX ACL, мапінг користувачів, узгодження SMB і NFS, тестування продуктивності та безпеки. У випадку використання готового NAS-рішення ці етапи були б частково приховані за графічним інтерфейсом.

Недоліком ручної конфігурації є вища складність адміністрування. Необхідно правильно налаштувати мережу, користувачів, групи, UID/GID, Samba-паролі, NFS-експорт, ACL, автоматичне монтування та служби. Помилка в одному з цих компонентів може призвести до того, що користувач не зможе підключитися до ресурсу або отримає неправильні права доступу. Проте для навчальної роботи ця складність є не недоліком, а важливою частиною дослідження, оскільки дозволяє глибше зрозуміти механізми роботи файлового сервера.

У межах цієї роботи було обрано саме підхід Linux + Samba/NFS. Він є найбільш доцільним, оскільки відповідає темі кваліфікаційної роботи, дозволяє реалізувати файловий сервер змішаного доступу на базі Linux і забезпечує можливість детального тестування. У ролі серверної операційної системи використовується Ubuntu Server, для Windows-доступу — Samba, для Linux-доступу — NFS, а для розмежування прав — POSIX ACL.

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		29

Таблиця 1.3

Порівняння підходів до побудови файлового сервера змішаного доступу

Критерій порівняння	TrueNAS	Готові NAS-рішення	Linux + Samba/NFS
Тип рішення	Спеціалізована NAS-платформа	Апаратно-програмний NAS-пристрій	Самостійно налаштований Linux-сервер
Вартість	Безкоштовна система, але потрібне обладнання	Залежить від моделі, часто висока	Мінімальна, можливе використання віртуальних машин
Складність початкового налаштування	Середня	Низька	Середня або висока
Гнучкість конфігурації	Висока, але в межах платформи	Обмежена виробником	Максимальна
Наявність веб-інтерфейсу	Так	Так	Зазвичай ні, налаштування через CLI
Підтримка SMB	Так	Так	Так, через Samba
Підтримка NFS	Так	Так	Так, через nfs-kernel-server
Контроль над конфігураційними файлами	Частковий	Обмежений	Повний
Зручність для швидкого впровадження	Висока	Дуже висока	Середня
Основний недолік	Надмірність для простого стенду	Вартість і залежність від виробника	Потреба в ручному адмініструванні

З аналізу видно, що кожен підхід має свою сферу застосування. TrueNAS і готові NAS-пристрої зручні тоді, коли потрібне швидке розгортання сховища з мінімальним ручним налаштуванням. Водночас ручне розгортання Linux-сервера з Samba та NFS краще підходить для дослідницької та навчальної роботи, оскільки забезпечує повний контроль над усіма компонентами системи.

Таким чином, для реалізації теми «Проект файлового сервера змішаного доступу SMB/NFS на базі Linux» найбільш обґрунтованим є вибір Linux-сервера з ручним налаштуванням Samba та NFS. Це дозволяє побудувати гнучке рішення,

адаптоване до вимог лабораторного стенду, і водночас дослідити основні технічні аспекти змішаного файлового доступу.

Висновки до розділу 1

У першому розділі було виконано аналіз предметної області побудови файлових серверів змішаного доступу в гетерогенних комп'ютерних мережах. Розглянуто призначення файлових серверів як важливих елементів мережевої інфраструктури, що забезпечують централізоване зберігання даних, спільну роботу користувачів, контроль прав доступу, спрощення адміністрування та підвищення надійності роботи з файлами.

Встановлено, що в середовищах, де одночасно використовуються Windows- та Linux-клієнти, доцільним є застосування двох протоколів файлового доступу: SMB/CIFS та NFS. Протокол SMB/CIFS є типовим для Windows-середовища і на Linux-сервері реалізується за допомогою програмного комплексу Samba. Протокол NFS є природним рішенням для Linux- та Unix-подібних систем і дозволяє монтувати віддалені серверні каталоги у локальну файлову систему клієнта. Поєднання цих протоколів дає змогу створити єдиний спільний файловий ресурс для різних операційних систем.

Проаналізовано проблематику одночасного змішаного доступу до одного серверного каталогу через SMB і NFS. Визначено, що основними технічними ризиками такого підходу є конфлікти блокування файлів, особливості кешування, затримки відображення змін, можливість перезапису даних та різна поведінка клієнтів Windows і Linux під час роботи з одним і тим самим файлом. Для зменшення цих ризиків необхідно правильно налаштовувати права доступу, використовувати POSIX ACL, контролювати успадкування прав для нових файлів і визначати правила одночасної роботи користувачів зі спільними даними.

Розглянуто відмінності між моделями прав доступу POSIX ACL та Windows ACL. Показано, що Linux використовує UID, GID, стандартні POSIX-права та POSIX ACL, тоді як Windows базується на SID і власній ACL-моделі. У файловому сервері змішаного доступу на базі Linux центральною є модель прав Linux-

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		31

файлової системи, а Samba виконує роль проміжного рівня, який дозволяє Windows-клієнтам працювати з файловим ресурсом. Для NFS важливим є узгодження UID/GID між сервером і Linux-клієнтами.

Проведено огляд можливих аналогів і підходів до побудови файлового сервера змішаного доступу: TrueNAS, готових NAS-рішень і самостійної Linux-конфігурації з Samba та NFS. За результатами порівняння встановлено, що готові NAS-платформи є зручними для швидкого впровадження, однак частково приховують внутрішні механізми роботи файлових служб. Натомість ручне розгортання Linux-сервера забезпечує повний контроль над конфігурацією, високу навчальну цінність і можливість детально дослідити роботу SMB, NFS, POSIX ACL та механізмів автентифікації.

Отже, за результатами аналізу предметної області сформовано постановку задачі: необхідно спроектувати файловий сервер змішаного доступу SMB/NFS на базі Linux, який забезпечує доступ Windows-клієнтів через Samba, Linux-клієнтів через NFS, використовує єдиний спільний каталог, підтримує розмежування прав доступу на основі користувачів, груп і POSIX ACL, а також проходить функціональне, продуктивне та безпекове тестування.

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		32

РОЗДІЛ 2 МОДЕЛЮВАННЯ ФАЙЛОВОГО СЕРВЕРА ЗМІШАНОГО ДОСТУПУ

2.1 Обґрунтування вибору дистрибутива Linux і файлової системи (ext4/XFS/ZFS); організація дискового простору

Для реалізації файлового сервера змішаного доступу SMB/NFS необхідно обрати серверну операційну систему, яка забезпечує стабільну роботу мережеслужб, підтримку сучасних файлових систем, можливість гнучкого налаштування користувачів і груп, підтримку POSIX ACL, а також сумісність із Samba та NFS. У межах цієї кваліфікаційної роботи як серверну платформу обрано Linux-дистрибутив Ubuntu Server.

Ubuntu Server є доцільним вибором для побудови лабораторного файлового сервера з кількох причин. По-перше, цей дистрибутив має широку підтримку спільноти та велику кількість документації, що спрощує процес розгортання і налагодження системи. По-друге, Ubuntu Server підтримує всі необхідні компоненти для реалізації змішаного файлового доступу: Samba, NFS Server, OpenSSH Server, утиліти керування користувачами, групами, правами доступу та ACL. По-третє, система є безкоштовною, відкритою та достатньо стабільною для використання в навчальному, лабораторному або малому офісному середовищі.

Важливою перевагою Ubuntu Server є зручність адміністрування через командний рядок. Для файлового сервера це має суттєве значення, оскільки більшість налаштувань виконується через конфігураційні файли та системні служби. Наприклад, параметри Samba задаються у файлі `/etc/samba/smb.conf`, параметри NFS-експорту — у файлі `/etc/exports`, а автоматичне монтування NFS-ресурсу на Linux-клієнті — у файлі `/etc/fstab`. Такий підхід дозволяє чітко контролювати конфігурацію та відтворювати її під час документування роботи.

Крім Ubuntu Server, для побудови файлового сервера можна було б використати інші Linux-дистрибутиви, наприклад Debian, Rocky Linux, AlmaLinux або openSUSE. Debian має високу стабільність і є близьким до Ubuntu за структурою пакетів. Rocky Linux та AlmaLinux орієнтовані на серверне

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		33

використання і сумісні з екосистемою Red Hat. Проте для лабораторного стенду Ubuntu Server є зручнішим варіантом завдяки простішому встановленню, широкій підтримці пакетів і зрозумілій документації.

Окрему увагу під час моделювання файлового сервера необхідно приділити вибору файлової системи. Файлова система визначає спосіб організації даних на диску, підтримку прав доступу, журналювання, роботу з великими файлами, відновлення після збоїв і загальну продуктивність. Для файлових серверів на базі Linux найчастіше розглядають ext4, XFS та ZFS.

Файлова система ext4 є однією з найпоширеніших у Linux-середовищі. Вона використовується за замовчуванням у багатьох дистрибутивах, має стабільну реалізацію, підтримує журналювання, POSIX-права, POSIX ACL, символічні посилання та стандартні Linux-механізми керування файлами. Ext4 добре підходить для універсальних серверних завдань, зокрема для лабораторного файлового сервера, де важлива стабільність, простота адміністрування та сумісність із Samba і NFS.

XFS є високопродуктивною журналювальною файловою системою, яка добре працює з великими файлами та великими обсягами даних. Вона часто використовується на серверах, де потрібна висока швидкість роботи з великими файлами, наприклад у системах резервного копіювання, медіасховищах або сховищах великих архівів. Проте для невеликого лабораторного стенду переваги XFS не є критичними, а ext4 є простішою для використання та достатньою за можливостями.

ZFS є більш складною файловою системою та водночас менеджером томів. Вона підтримує знімки, перевірку цілісності даних, дедуплікацію, стиснення, пули дисків та інші розширені можливості. ZFS часто використовується у NAS-рішеннях і системах, де важливо забезпечити високу надійність зберігання даних. Однак ZFS потребує більше ресурсів, уважнішого планування та складнішого адміністрування. Для теми цієї кваліфікаційної роботи, де головна увага приділяється саме змішаному доступу SMB/NFS, використання ZFS є не обов'язковим.

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		34

Таблиця 2.1

Порівняння файлових систем ext4, XFS та ZFS

Критерій порівняння	ext4	XFS	ZFS
Тип файлової системи	Журналювальна Linux-файлова система	Журналювальна файлова система для великих обсягів даних	Файлова система з функціями менеджера томів
Складність адміністрування	Низька	Середня	Висока
Підтримка POSIX ACL	Так	Так	Так
Робота з великими файлами	Добра	Дуже добра	Дуже добра
Вбудована перевірка цілісності	Обмежена	Обмежена	Розширена
Підтримка знімків	Ні, без додаткових засобів	Ні, без додаткових засобів	Так
Доцільність для лабораторного стенду	Найбільш доцільна	Можлива, але не обов'язкова	Надмірна для базового стенду
Основна перевага	Простота, стабільність, сумісність	Продуктивність із великими файлами	Надійність, знімки, контроль цілісності
Основний недолік	Менше розширених можливостей	Менш гнучка для зменшення розділів	Складність і більша ресурсомісткість

За результатами порівняння для практичної реалізації було обрано файлову систему ext4. Такий вибір пояснюється тим, що ext4 є стабільною, простою в адмініструванні, добре підтримується Ubuntu Server і повністю сумісна з необхідними механізмами доступу: POSIX-правами, POSIX ACL, Samba та NFS. Для лабораторного стенду, побудованого у віртуальному середовищі, ext4 є достатньою і не створює зайвої складності під час налаштування.

Організація дискового простору файлового сервера повинна бути зрозумілою, логічною та придатною для подальшого адміністрування. У Linux для серверних даних часто використовують каталог /srv, призначений для розміщення даних сервісів, які надаються системою. Саме тому в межах цієї роботи для файлового ресурсу використано шлях /srv/storage/mixed.

Каталог `/srv/storage` виконує роль базового каталогу для файлового сховища, а `/srv/storage/mixed` є безпосередньо спільним каталогом, який надається клієнтам через SMB і NFS. Такий підхід дозволяє відокремити системні файли операційної системи від даних файлового сервера. У майбутньому, за необхідності, каталог `/srv/storage` можна перенести на окремий диск або окремий розділ без зміни логіки роботи Samba та NFS.

У спільному каталозі доцільно створити кілька підкаталогів для впорядкування даних. Наприклад, каталог `public` може використовуватися для загальнодоступних файлів, `projects` — для робочих матеріалів, `archive` — для архівних або застарілих документів. У межах лабораторного стенду також створювалися тестові файли для перевірки доступу з Windows- і Linux-клієнтів. Наявність таких файлів дозволяє підтвердити, що один і той самий каталог доступний через обидва протоколи.

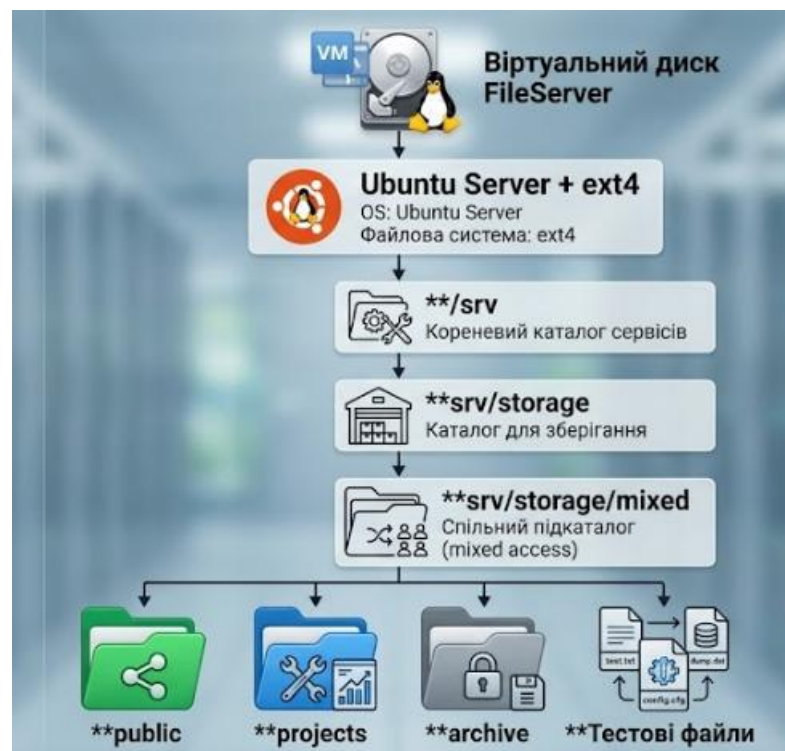


Рисунок 2.1 — Організація дискового простору

Узагальнена структура дискового простору файлового сервера має такий вигляд:

```
/srv
/srv/storage
/srv/storage/mixed
/srv/storage/mixed/public
/srv/storage/mixed/projects
/srv/storage/mixed/archive
```

Така структура є простою, але достатньою для демонстрації роботи файлового сервера змішаного доступу. Вона дозволяє реалізувати єдиний файловий ресурс, який одночасно публікується через Samba та експортується через NFS. Крім того, вона зручна для подальшого розширення, оскільки за потреби можна створювати додаткові каталоги для окремих відділів, груп користувачів або типів даних.

Отже, для моделювання файлового сервера змішаного доступу обрано Ubuntu Server як стабільну та зручну серверну платформу, а файлову систему ext4 — як просте, надійне і сумісне рішення для лабораторного стенду. Дисковий простір організовано навколо каталогу /srv/storage/mixed, який є центральним спільним ресурсом для Windows- і Linux-клієнтів.

2.2 Проєктування архітектури системи (структурна схема компонентів: ОС, дискова підсистема, Samba, NFS-сервер, служба автентифікації)

Проєктування архітектури файлового сервера змішаного доступу є важливим етапом, оскільки саме на цьому рівні визначаються основні компоненти системи, їх призначення, взаємодія між собою та роль кожного вузла в лабораторному стенді. Архітектура повинна забезпечувати доступ до спільного файлового ресурсу з боку Windows- та Linux-клієнтів, підтримувати розмежування прав доступу, бути простою для адміністрування та придатною для подальшого функціонального і продуктивного тестування.

У межах кваліфікаційної роботи було спроектовано лабораторний стенд, що складається з трьох основних віртуальних машин: FileServer, UbuntuClient та Windows VM. Такий підхід дозволяє змодельовати гетерогенну мережу, у якій один

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				37
Змн.	Арк.	№ докум.	Підпис	Дата		

Linux-сервер одночасно обслуговує клієнтів різних операційних систем. Віртуалізація дає можливість ізолювати середовище, швидко змінювати налаштування, перевіряти роботу мережеслужб і відновлювати систему після помилок.

Центральним елементом архітектури є FileServer — віртуальна машина на базі Ubuntu Server. Саме на цьому вузлі розміщено спільний файловий каталог, встановлено та налаштовано служби Samba і NFS, створено користувачів і групи, а також визначено права доступу за допомогою POSIX ACL. FileServer виконує роль єдиного сховища даних, до якого клієнти підключаються різними протоколами.

Windows VM використовується як клієнтська система для перевірки доступу через SMB/CIFS. Для Windows-клієнта спільний ресурс надається у вигляді мережевої папки \\192.168.56.10\\mixed або мережевого диска. Доступ до ресурсу здійснюється через Samba, яка приймає SMB-запити, виконує автентифікацію користувача та передає операції читання або запису до Linux-файлової системи.

UbuntuClient використовується як Linux-клієнт для перевірки доступу через NFS. На цьому вузлі серверний каталог монтується в локальну файлову систему як /mnt/mixed. Після монтування користувач може працювати з віддаленими файлами за допомогою звичайних Linux-команд, наприклад ls, cat, touch, echo, rm. Такий сценарій дозволяє перевірити роботу NFS, коректність UID/GID та застосування прав доступу на рівні Linux.

Для взаємодії між вузлами використовується ізольована Host-only мережа з адресним простором 192.168.56.0/24. Такий тип мережі є зручним для лабораторного стенду, оскільки дозволяє віртуальним машинам обмінюватися даними між собою та з основною системою, але не відкриває стенд безпосередньо в зовнішню мережу. Це зменшує ризики під час тестування служб SSH, SMB і NFS.

У розробленій архітектурі FileServer має IP-адресу 192.168.56.10, UbuntuClient — 192.168.56.20, а Windows VM — 192.168.56.30. Така адресація є простою та логічною: сервер отримує першу адресу в обраній групі, а клієнти — наступні адреси. Це спрощує тестування, оскільки всі команди підключення використовують одну стабільну адресу файлового сервера.

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		38

Таблиця 2.2

Адресація вузлів лабораторного стенду

Вузол	Операційна система	Роль у стенді	IP-адреса	Основне призначення
FileServ	Ubuntu Server	Сервер	192.168.56.10	Розміщення спільного каталогу, Samba, NFS, SSH
UbuntuClient	Ubuntu Desktop	Linux-клієнт	192.168.56.20	Доступ до спільного каталогу через NFS
Win10Client	Windows	Windows-клієнт	192.168.56.30	Доступ до спільного каталогу через SMB/CIFS

Операційна система FileServer виконує роль базової платформи для всіх серверних служб. На ній встановлюються пакети Samba, nfs-kernel-server, openssh-server, acl та допоміжні утиліти для тестування. Ubuntu Server працює без надмірного графічного інтерфейсу, що зменшує використання ресурсів і робить систему ближчою до реального серверного середовища.

Дискова підсистема FileServer організована навколо файлової системи ext4. У структурі каталогів створено окрему область /srv/storage, у якій розміщено спільний каталог /srv/storage/mixed. Саме цей каталог є центральним ресурсом системи. Він одночасно публікується через Samba та експортується через NFS. Таким чином, Windows- і Linux-клієнти працюють з одним і тим самим фізичним каталогом на сервері, але використовують різні протоколи доступу.

Служба Samba є компонентом, який забезпечує доступ Windows-клієнтів до Linux-сервера. У конфігураційному файлі /etc/samba/smb.conf створюється ресурс mixed, у якому задається шлях до каталогу /srv/storage/mixed, дозволені користувачі, режими доступу та параметри створення файлів. Samba також використовує локальні облікові записи користувачів, які додаються до Samba-бази за допомогою команди smbpasswd.

NFS-сервер забезпечує доступ Linux-клієнтів до того самого каталогу. Для цього в конфігураційному файлі /etc/exports задається експорт каталогу /srv/storage/mixed для мережі 192.168.56.0/24. Після застосування налаштувань клієнт UbuntuClient може змонтувати цей ресурс у каталог /mnt/mixed. У практичній реалізації також було налаштовано автоматичне монтування через /etc/fstab, щоб ресурс підключався після перезавантаження клієнтської системи.

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				39
Змн.	Арк.	№ докум.	Підпис	Дата		

Служба автентифікації в межах лабораторного стенду реалізована на основі локальних користувачів Linux і Samba-облікових записів. Для демонстрації розмежування доступу було створено користувачів ivan і petro. Користувач ivan має права читання та запису, а користувач petro — лише читання. Такий підхід дозволяє перевірити, що файлова система правильно обмежує дії користувачів незалежно від того, через який протокол вони підключаються.

Окрему роль в архітектурі відіграють групи доступу. Для керування правами було передбачено групи fs_data, fs_rw, fs_ro та fs_admins. Група fs_rw використовується для користувачів із правом запису, fs_ro — для користувачів із правом лише читання, fs_admins — для адміністраторів, а fs_data — як загальна група для файлового ресурсу. Така модель дозволяє не призначати права кожному користувачеві окремо, а керувати доступом через групи.

POSIX ACL використовується як механізм детального налаштування прав доступу до спільного каталогу. Завдяки ACL можна одночасно надати різні права кільком групам. Наприклад, група fs_rw отримує права rwx, група fs_ro — права r-x, а група fs_admins — права rwx. Також налаштовуються default ACL, які забезпечують успадкування прав для нових файлів і каталогів, створених у спільному ресурсі.

Структурна схема компонентів системи наведена на рисунку 2.2.

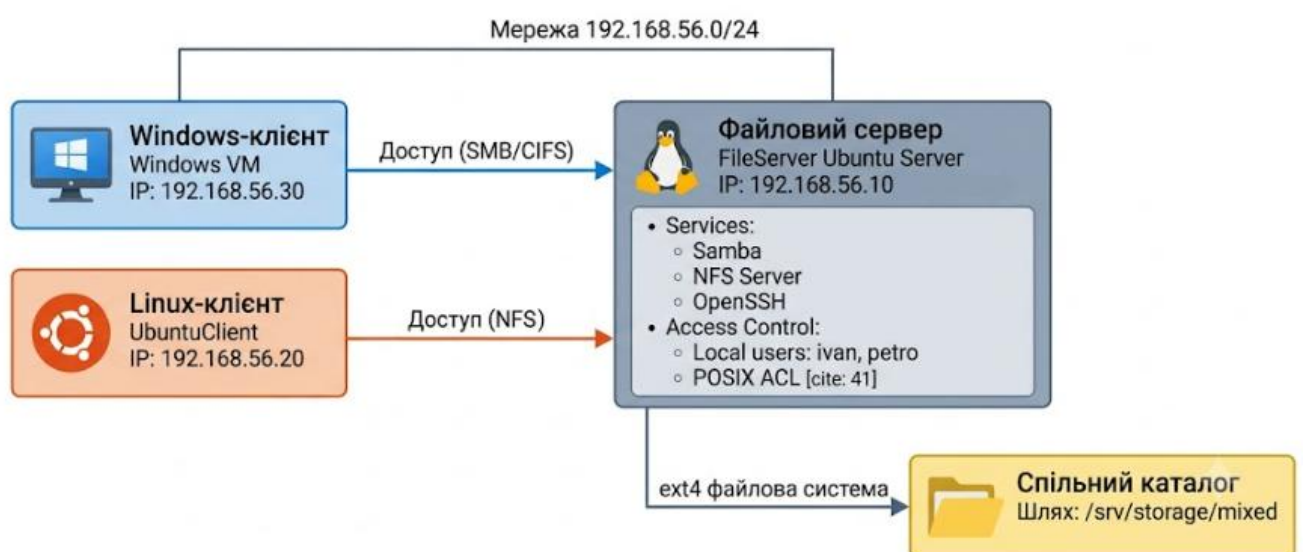


Рисунок 2.2 — Структурна схема файлового сервера змішаного доступу SMB/NFS

На схемі відображено три основні рівні: клієнтський рівень, серверний рівень і рівень зберігання даних. До клієнтського рівня належать Windows VM і UbuntuClient. До серверного рівня належить FileServer із встановленими службами Samba, NFS Server, OpenSSH та локальною автентифікацією. До рівня зберігання даних належить файлова система ext4 і каталог /srv/storage/mixed.

Архітектура системи передбачає таку логіку взаємодії. Windows-клієнт надсилає SMB-запит до Samba-сервера, проходить автентифікацію як користувач ivan або petro, після чого Samba звертається до каталогу /srv/storage/mixed і виконує операцію відповідно до прав Linux-файлової системи. Linux-клієнт монтує той самий каталог через NFS і виконує файлові операції безпосередньо через механізми NFS та POSIX-прав. Усі операції в кінцевому результаті контролюються правами на серверному каталозі.

Запропонована архітектура є достатньою для моделювання файлового сервера змішаного доступу в лабораторному середовищі. Вона дозволяє перевірити роботу SMB і NFS, протестувати доступ із різних операційних систем, дослідити розмежування прав користувачів, виконати тестування продуктивності та оцінити базові аспекти безпеки. Крім того, така архітектура може бути розширена в майбутньому шляхом додавання централізованої автентифікації через LDAP, Active Directory або FreeIPA.

2.3 Модель прав доступу та автентифікації: інтеграція з LDAP/Active Directory/FreeIPA, узгодження ACL між протоколами

Модель прав доступу та автентифікації є одним із центральних елементів файлового сервера змішаного доступу SMB/NFS. Саме вона визначає, які користувачі можуть підключатися до файлового ресурсу, які операції їм дозволено виконувати та яким чином права застосовуються під час доступу через різні протоколи. У системі змішаного доступу це питання є особливо важливим, оскільки Windows-клієнти працюють через SMB/Samba, а Linux-клієнти — через NFS, тоді як фактичне зберігання файлів відбувається на Linux-файловій системі.

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		41

У файлового сервері на базі Linux основою контролю доступу є користувачі, групи, UID, GID, стандартні POSIX-права та POSIX ACL. Коли клієнт виконує операцію читання, запису або видалення файлу, серверна операційна система перевіряє, чи має відповідний користувач або група достатні права для виконання цієї дії. Тому навіть якщо доступ здійснюється через Samba або NFS, кінцеве рішення щодо дозволу або заборони операції приймається на рівні Linux-файлової системи.

Для SMB-доступу використовується Samba, яка виконує автентифікацію користувачів Windows-клієнтів і зіставляє їх із локальними Linux-користувачами. Наприклад, користувач Windows підключається до ресурсу \\192.168.56.10\\mixed під обліковим записом ivan. Samba перевіряє наявність такого користувача у власній базі облікових записів і після успішної автентифікації виконує файлові операції від імені відповідного Linux-користувача ivan. Таким чином, доступ через SMB фактично залежить не лише від параметрів smb.conf, а й від прав Linux-користувача на каталог /srv/storage/mixed.

Для NFS-доступу використовується інший принцип. NFS-клієнт не проходить автентифікацію в такому самому вигляді, як SMB-клієнт. Натомість важливу роль відіграють числові ідентифікатори UID і GID. Якщо користувач ivan на сервері має UID 1101, то на Linux-клієнті бажано створити користувача ivan із таким самим UID. У такому разі сервер і клієнт однаково інтерпретують власника файлів і права доступу. Якщо UID/GID не узгоджені, можуть виникнути помилки, коли користувач бачить файл як такий, що належить іншому власнику, або не має очікуваних прав доступу.

У великих організаціях для вирішення проблеми єдиної автентифікації та узгодження користувачів застосовують централізовані служби каталогів. Найпоширенішими варіантами є LDAP, Microsoft Active Directory та FreeIPA. Вони дозволяють зберігати користувачів і групи в єдиному каталозі, до якого звертаються сервери та клієнтські системи. У такому випадку адміністратор створює користувача один раз у централізованій службі, а всі підключені системи використовують єдину базу ідентифікації.

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		42

Active Directory є типовим рішенням для Windows-орієнтованих корпоративних мереж. Вона забезпечує централізоване керування користувачами, групами, політиками безпеки, доменними комп'ютерами та правами доступу. Samba може бути інтегрована з Active Directory, що дозволяє Linux-серверу працювати у складі доменної інфраструктури та використовувати доменні облікові записи для доступу до SMB-ресурсів. Такий варіант є зручним для великих організацій, де вже існує Windows-домен.

LDAP є універсальним протоколом доступу до служб каталогів. Він може використовуватися для централізованого зберігання облікових записів користувачів і груп у Linux- та змішаних середовищах. На основі LDAP можна побудувати єдину систему ідентифікації для серверів і клієнтів, що зменшує потребу в ручному створенні однакових користувачів на кожному вузлі мережі. Проте LDAP потребує додаткового налаштування, адміністрування та захисту.

FreeIPA є комплексним рішенням для Linux/Unix-середовищ, яке поєднує LDAP, Kerberos, DNS, сертифікати та засоби керування політиками. FreeIPA може використовуватися для централізованої автентифікації Linux-клієнтів, узгодження UID/GID, керування групами та інтеграції з іншими службами. Для складних корпоративних середовищ це зручне рішення, однак для лабораторного стенду воно є надмірним за складністю.

У межах цієї кваліфікаційної роботи було обрано локальну модель автентифікації. Це означає, що користувачі та групи створюються безпосередньо на FileServer і, за потреби, на UbuntuClient з узгодженими UID/GID. Такий підхід є достатнім для лабораторного стенду, оскільки дозволяє наочно продемонструвати роботу Samba, NFS і POSIX ACL без додаткового розгортання доменної або каталогової інфраструктури. Інтеграція з LDAP, Active Directory або FreeIPA розглядається як можливий напрям подальшого розвитку та масштабування системи.

У практичній моделі доступу було передбачено кілька типів користувачів. Користувач іvan має права читання і запису до спільного каталогу. Він використовується для перевірки створення файлів через SMB і NFS. Користувач

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				43
Змн.	Арк.	№ докум.	Підпис	Дата		

petro має лише права читання. Він використовується для перевірки обмеження доступу та підтвердження того, що користувач без прав запису не може створювати або змінювати файли. Адміністративний користувач використовується для налаштування служб, зміни конфігураційних файлів і керування системою.

Для групового розмежування доступу було передбачено кілька груп. Група fs_rw призначена для користувачів, які мають право читання та запису. Група fs_ro використовується для користувачів, яким дозволено лише читання. Група fs_admins призначена для адміністраторів файлового ресурсу. Група fs_data може використовуватися як загальна група, пов'язана зі спільним файловим каталогом. Такий поділ дозволяє керувати правами доступу через членство в групах, що є зручнішим, ніж призначення прав кожному користувачеві окремо.

Таблиця 2.3

Модель користувачів, груп і прав доступу

Користувач / група	Роль у системі	Рівень доступу	Призначення
administrator	Адміністратор системи	Повний адміністративний доступ	Налаштування ОС, Samba, NFS, ACL, служб
ivan	Звичайний користувач із правом запису	Читання і запис	Перевірка створення та зміни файлів через SMB/NFS
petro	Користувач лише для читання	Тільки читання	Перевірка обмеження запису
fs_rw	Група користувачів із правом запису	gwx	Надання доступу користувачам, які можуть змінювати файли
fs_ro	Група користувачів лише для читання	r-x	Надання доступу користувачам, які можуть лише переглядати файли
fs_admins	Група адміністраторів ресурсу	gwx	Адміністрування спільного файлового каталогу
fs_data	Загальна група файлового ресурсу	Залежить від ACL	Логічне об'єднання доступу до файлових даних

Основним об'єктом контролю доступу є каталог /srv/storage/mixed. Саме він публікується через Samba і експортується через NFS. Для цього каталогу необхідно задати такі права, щоб вони однаково коректно працювали для Windows- і Linux-

клієнтів. Звичайних POSIX-прав у цьому випадку недостатньо, оскільки потрібно надати різні права кільком групам. Тому використовується POSIX ACL.

POSIX ACL дозволяють призначити окремі права для груп `fs_rw`, `fs_ro` і `fs_admins`. Група `fs_rw` отримує права `rwX`, що дозволяє читати, створювати, змінювати й видаляти файли відповідно до політики доступу. Група `fs_ro` отримує права `r-X`, тобто може переглядати каталог і читати файли, але не може створювати або змінювати об'єкти. Група `fs_admins` отримує права `rwX` для адміністрування ресурсу. Такий підхід дозволяє забезпечити розмежування доступу на рівні серверної файлової системи.

Важливим елементом є налаштування `default ACL`. Якщо звичайні ACL визначають права для вже існуючих файлів і каталогів, то `default ACL` задають права, які автоматично успадковуються новими об'єктами. Це особливо важливо для змішаного доступу SMB/NFS. Наприклад, якщо користувач `ivan` створює файл через SMB, цей файл повинен залишатися доступним для Linux-клієнта через NFS відповідно до групових прав. Без `default ACL` нові файли можуть отримувати надто обмежені права, що порушить логіку спільного доступу.

Узгодження ACL між протоколами полягає в тому, що Samba і NFS мають працювати з одним і тим самим набором серверних прав. Samba може мати власні параметри доступу, наприклад `valid users`, `read only`, `write list`, `create mask` і `directory mask`, але ці параметри не скасовують Linux-права. Якщо Linux-файлова система забороняє запис, Samba також не зможе дозволити користувачеві записати файл. Аналогічно NFS-доступ також обмежується правами Linux-файлової системи. Тому POSIX ACL є центральним механізмом узгодження доступу між SMB і NFS.

У моделі доступу можна виділити три рівні перевірки. Перший рівень — автентифікація користувача. Для SMB вона виконується засобами Samba, для NFS — через відповідність UID/GID. Другий рівень — членство користувача у відповідних групах, наприклад `fs_rw` або `fs_ro`. Третій рівень — перевірка POSIX ACL на каталозі або файлі. Лише якщо всі рівні дозволяють операцію, користувач може виконати дію з файлом.

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		45

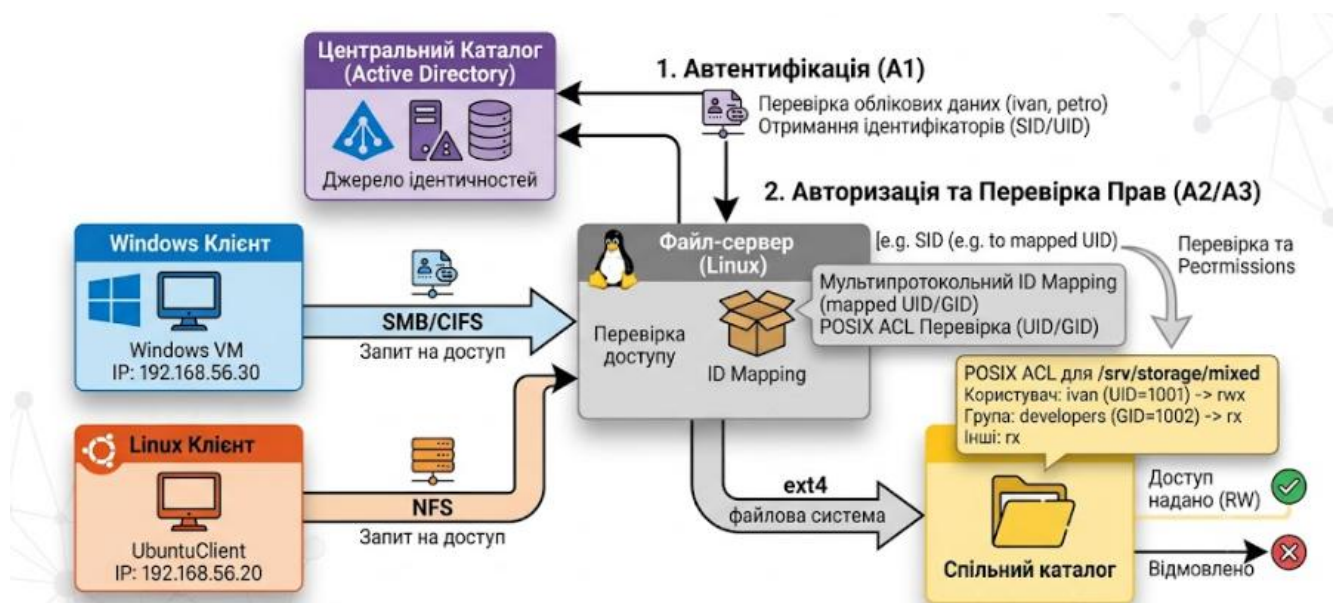


Рисунок 2.3 — Модель автентифікації та контролю доступу до спільного каталогу

Для користувача ivan модель працює таким чином: він автентифікується через Samba або працює через NFS з відповідним UID, належить до групи fs_rw і отримує права на читання та запис. У результаті він може створювати файли в каталозі /srv/storage/mixed, а ці файли стають видимими і для Windows-, і для Linux-клієнта. Саме цей сценарій використовується для перевірки повноцінного доступу.

Для користувача petro модель працює інакше. Він належить до групи fs_ro і має лише права читання. Це означає, що він може переглядати файли та каталоги, але не може створювати нові файли або змінювати наявні. Така поведінка перевіряється окремо через SMB і NFS. Якщо під час тесту користувач petro отримує повідомлення Permission denied або Access is denied, це підтверджує коректність моделі обмеження прав.

Окремо варто враховувати механізм root_squash у NFS. Його призначення полягає в тому, щоб користувач root на клієнтській машині не отримував автоматично root-права на серверному файловому ресурсі. Це важливий механізм безпеки, оскільки без нього адміністратор клієнтської машини міг би отримати надмірний доступ до файлів на сервері. У практичній частині роботи робота цього механізму була підтверджена під час спроби виконати операції з файлами через sudo на NFS-клієнті.

Отже, модель прав доступу та автентифікації в цій роботі базується на локальних Linux-користувачах, Samba-облікових записах, узгодженні UID/GID для

NFS, груповому розмежуванні доступу та POSIX ACL. Такий підхід є достатнім для лабораторного стенду та дозволяє перевірити основні принципи роботи файлового сервера змішаного доступу. Для корпоративного середовища запропонована модель може бути розширена шляхом інтеграції з LDAP, Active Directory або FreeIPA.

2.4 Сценарії використання та діаграми взаємодії клієнтів із сервером

Сценарії використання дозволяють формалізувати, яким чином клієнти взаємодіють із сервером, які дії виконують користувачі, які служби беруть участь в обробці запитів і на якому етапі перевіряються права доступу. Для файлового сервера SMB/NFS основними є сценарії доступу з Windows-клієнта через SMB, доступу з Linux-клієнта через NFS, перевірки прав користувачів і сценарій змішаного доступу до одного спільного каталогу.

Перший сценарій передбачає доступ Windows-клієнта до файлового ресурсу через протокол SMB/CIFS. У цьому випадку користувач на Windows VM відкриває мережевий шлях \\192.168.56.10\\mixed або підключає його як мережевий диск. Запит надсилається на FileServer до служби Samba. Samba приймає SMB-запит, перевіряє облікові дані користувача, визначає відповідний Linux-обліковий запис і звертається до файлової системи сервера. Якщо користувач має достатні права згідно з POSIX ACL, операція читання або запису виконується.

У межах практичної реалізації Windows-клієнт використовує обліковий запис іvan для перевірки повного доступу до ресурсу. Користувач іvan має право створювати файли, редагувати їх і переглядати вміст каталогу. Також використовується обліковий запис retro для перевірки обмеженого доступу. У цьому випадку користувач може переглядати вміст ресурсу, але не може створювати нові файли або змінювати наявні. Такий сценарій дозволяє перевірити, чи правильно Samba взаємодіє з Linux-правами доступу.

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		47



Рисунок 2.4 — Сценарій доступу Windows-клієнта до спільного каталогу через SMB

Послідовність взаємодії Windows-клієнта із сервером через SMB:

- Користувач Windows відкриває ресурс \192.168.56.10\mixed.
- Windows-клієнт надсилає SMB-запит до FileServer.
- Служба Samba запитує облікові дані користувача.
- Samba перевіряє користувача у власній базі облікових записів.
- Користувач зіставляється з локальним Linux-користувачем.
- Linux-файлова система перевіряє POSIX ACL для каталогу /srv/storage/mixed.
- Якщо права дозволяють операцію, користувач отримує доступ до файлів.
- Якщо прав недостатньо, клієнт отримує повідомлення про відмову в доступі.

Другий сценарій передбачає доступ Linux-клієнта до того самого спільного каталогу через NFS. У цьому випадку UbuntuClient монтує віддалений каталог FileServer у локальну директорію /mnt/mixed. Після монтування користувач може

працювати з ресурсом за допомогою стандартних Linux-команд. Наприклад, команда `ls` дозволяє переглянути вміст каталогу, `touch` — створити файл, `echo` — записати текст, `cat` — переглянути вміст файлу.

На відміну від SMB, NFS орієнтується на UID і GID. Тому важливо, щоб користувачі на сервері та клієнті були узгоджені. Якщо користувач `ivan` має відповідний UID і входить до групи з правами запису, він може створювати файли в `/mnt/mixed`. Якщо користувач `petro` має лише права читання, спроба створення файлу повинна завершитися помилкою `Permission denied`. Така поведінка підтверджує правильність налаштування NFS і POSIX ACL.

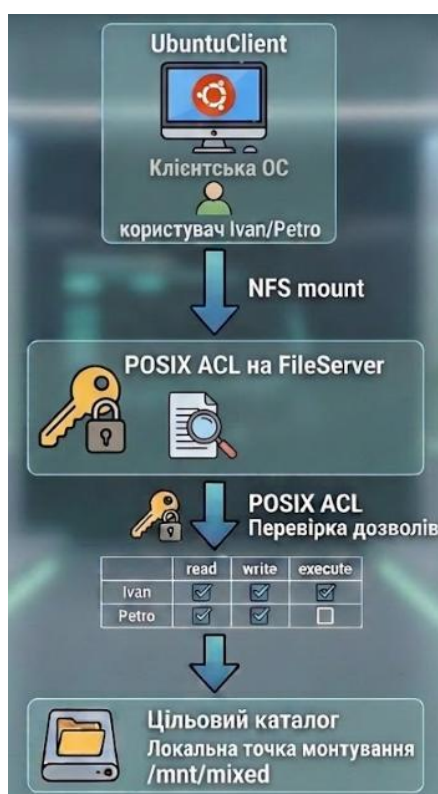


Рисунок 2.5 — Сценарій доступу Linux-клієнта до спільного каталогу через NFS

Послідовність взаємодії Linux-клієнта із сервером через NFS:

- UbuntuClient звертається до NFS-ресурсу `192.168.56.10:/srv/storage/mixed`.
- Сервер перевіряє, чи дозволено експорт для мережі `192.168.56.0/24`.
- Віддалений каталог монтується в локальну точку `/mnt/mixed`.
- Користувач Linux виконує файлову операцію.
- NFS передає запит на FileServer.
- FileServer перевіряє UID/GID користувача та POSIX ACL.

- Якщо права дозволяють дію, операція виконується.
- Якщо прав недостатньо, клієнт отримує повідомлення Permission denied.

Третій сценарій стосується змішаного доступу, коли один і той самий каталог використовується одночасно через SMB і NFS. Наприклад, Windows-клієнт створює файл windows_smb_test.txt у мережевій папці \\192.168.56.10\\mixed, після чого Linux-клієнт перевіряє наявність цього файлу в каталозі /mnt/mixed. У зворотному напрямку Linux-клієнт створює файл nfs_ivan_test.txt через NFS, після чого Windows-клієнт бачить його у SMB-ресурсі. Такий сценарій підтверджує, що обидва протоколи працюють з одним і тим самим серверним каталогом /srv/storage/mixed.

Четвертий сценарій пов'язаний із перевіркою прав доступу. Для користувача ivan очікуваним результатом є успішне створення і читання файлів через SMB та NFS. Для користувача petro очікуваним результатом є можливість читати файли, але неможливість створювати або змінювати їх. Якщо під час спроби запису користувач petro отримує повідомлення Access is denied у Windows або Permission denied у Linux, це свідчить про коректне застосування моделі прав доступу.

П'ятий сценарій стосується перевірки механізмів блокування та стану відкритих файлів. Якщо файл відкритий Windows-клієнтом через SMB, на сервері можна використати команду smbstatus --locks для перевірки активних SMB-блокувань або lease. Такий тест дозволяє зафіксувати, що Samba відстежує відкриті файли та стан доступу до них. Для змішаного SMB/NFS-доступу це важливо, оскільки дозволяє оцінити поведінку SMB-частини системи під час роботи з відкритими файлами.

Для систематизації сценаріїв використання сформовано таблицю 2.4.

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		50

Таблиця 2.4

Основні сценарії використання файлового сервера змішаного доступу

№	Сценарій	Клієнт	Протокол	Очікуваний результат
1	Підключення Windows-клієнта до ресурсу mixed	Windows VM	SMB/CIFS	Відкриття \192.168.56.10\mixed
2	Підключення Linux-клієнта до ресурсу mixed	UbuntuClient	NFS	Монтування /mnt/mixed
3	Створення файлу користувачем ivan	Windows VM / UbuntuClient	SMB/NFS	Файл створюється успішно
4	Спроба запису користувачем petro	Windows VM / UbuntuClient	SMB/NFS	Відмова в доступі
5	Перевірка змішаного доступу	Windows VM + UbuntuClient	SMB + NFS	Файл, створений одним клієнтом, видно іншому
6	Перевірка SMB-блокувань	Windows VM + FileServer	SMB	smbstatus показує відкриті файли або lease
7	Перевірка доступності сервера	Windows VM / UbuntuClient	SSH/SMB/NFS	Сервер відповідає та служби працюють
8	Перевірка автоматичного NFS-монтування	UbuntuClient	NFS	Після перезавантаження ресурс знову доступний

Описані сценарії використання покривають основні функціональні вимоги до файлового сервера змішаного доступу. Вони дозволяють перевірити підключення клієнтів, автентифікацію, права доступу, видимість файлів між SMB і NFS, обмеження для користувачів із правами лише читання та базову поведінку механізмів блокування. На основі цих сценаріїв у наступному підрозділі формується методика тестування системи.

2.5 Методика тестування системи

Методика тестування файлового сервера змішаного доступу SMB/NFS повинна охоплювати не лише перевірку факту підключення клієнтів до спільного каталогу, а й аналіз коректності прав доступу, працездатності обох протоколів, стабільності після перезавантаження, продуктивності передавання даних і базових механізмів безпеки. Оскільки система призначена для роботи в гетерогенній

мережі, тестування має виконуватися окремо з боку Windows-клієнта та Linux-клієнта, а також у сценарії змішаного доступу до одного спільного каталогу.

У межах кваліфікаційної роботи тестування системи поділено на кілька основних груп: перевірка мережевої доступності, функціональне тестування SMB і NFS, перевірка прав доступу користувачів, тестування змішаного доступу, перевірка механізмів блокування, навантажувальне тестування продуктивності, тестування безпеки та перевірка відмовостійкості після перезавантаження. Такий підхід дозволяє оцінити систему комплексно, а не лише підтвердити її базову працездатність.

Першим етапом тестування є перевірка мережевої доступності вузлів лабораторного стенду. Для цього використовуються команди `ping`, `ip -br` а та перевірка служб на сервері. На FileServer необхідно переконатися, що мережевий інтерфейс Host-only має IP-адресу 192.168.56.10. На UbuntuClient перевіряється адреса 192.168.56.20, а на Windows VM — адреса 192.168.56.30. Після цього клієнти повинні успішно виконувати `ping` до файлового сервера. Якщо сервер недоступний на мережевому рівні, подальше тестування SMB і NFS є неможливим.

Другим етапом є перевірка серверних служб. На FileServer необхідно перевірити стан служб `ssh`, `smbd` та `nfs-server`. Служба SSH потрібна для віддаленого адміністрування сервера, `smbd` забезпечує роботу Samba/SMB, а `nfs-server` відповідає за експорт NFS-ресурсів. Для перевірки використовуються команди `systemctl status ssh --no-pager`, `systemctl status smbd --no-pager` та `systemctl status nfs-server --no-pager`. Очікуваним результатом є активний стан відповідних служб.

Третім етапом є функціональне тестування SMB-доступу з Windows-клієнта. Windows VM повинна відкривати спільний ресурс за адресою `\192.168.56.10\mixed`. Після введення облікових даних користувача `ivan` має відкриватися вміст спільного каталогу. Додатково перевіряється можливість створення файлу через Windows-клієнт. Якщо файл успішно створюється, це означає, що Samba налаштована правильно, користувач автентифікується, а права доступу дозволяють запис.

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		52

Четвертим етапом є функціональне тестування NFS-доступу з Linux-клієнта. На UbuntuClient серверний каталог 192.168.56.10:/srv/storage/mixed монтується в локальну точку /mnt/mixed. Після монтування команда `mount | grep mixed` повинна показувати активне NFS-підключення. Команда `ls -la /mnt/mixed` повинна відображати вміст спільного каталогу. Для перевірки запису користувач іvan створює тестовий файл у /mnt/mixed. Якщо файл створюється без помилок, NFS-доступ працює коректно.

П'ятим етапом є перевірка змішаного доступу. Для цього файл, створений через SMB із Windows-клієнта, повинен бути видимим на UbuntuClient через NFS. Аналогічно файл, створений через NFS на UbuntuClient, повинен бути видимим у Windows VM через SMB. Такий тест підтверджує, що обидва протоколи працюють не з різними копіями даних, а з одним спільним серверним каталогом /srv/storage/mixed.

Шостим етапом є перевірка прав доступу користувачів. Для користувача іvan очікуваним результатом є можливість читання та запису. Для користувача petro очікуваним результатом є можливість читання, але неможливість створення або зміни файлів. У Windows-клієнті така перевірка виконується через підключення SMB-ресурсу під користувачем petro та спробу створення файлу. Очікуваним результатом є повідомлення Access is denied. На UbuntuClient аналогічна перевірка виконується командою `sudo -u petro touch /mnt/mixed/petro_security_test.txt`. Очікуваним результатом є повідомлення Permission denied.

Сьомим етапом є перевірка механізмів блокування та відкритих файлів у Samba. Для цього на Windows-клієнті відкривається файл зі спільного SMB-ресурсу, після чого на FileServer виконується команда `sudo smbstatus --locks`. Очікуваним результатом є відображення інформації про відкритий файл або SMB lease. Цей тест дозволяє підтвердити, що Samba відстежує відкриті SMB-файли та використовує механізми контролю доступу до них.

Восьмим етапом є навантажувальне тестування продуктивності. Воно включає окрему перевірку пропускнуої здатності мережі, швидкості запису та читання через NFS, а також швидкості запису й читання через SMB. Для

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		53

мережевого тестування використовується `iperf3`. На FileServer запускається `iperf3 -s`, а на UbuntuClient виконуються команди `iperf3 -c 192.168.56.10 -t 30` та `iperf3 -c 192.168.56.10 -t 30 -R`. Це дозволяє оцінити швидкість передавання даних у прямому та зворотному напрямках.

Для тестування NFS використовується утиліта `dd`. Запис перевіряється створенням тестового файлу в каталозі `/mnt/mixed`, а читання — зчитуванням цього файлу в `/dev/null`. Такий тест дозволяє оцінити практичну швидкість роботи з файлами через NFS. Для тестування SMB використовується `robocopy` на Windows VM. Спочатку створюється тестовий файл на локальному диску Windows, після чого він копіюється на мережевий диск SMB, а потім копіюється назад із SMB-ресурсу на локальний диск. Результати `robocopy` дозволяють оцінити швидкість запису та читання через SMB.

Дев'ятим етапом є тестування безпеки. Воно включає перевірку відкритих портів сервера за допомогою `ntmap`, перевірку журналювання невдалих SSH-спроб входу, перевірку обмежень користувача `petro` та перевірку механізму `root_squash` для NFS. Сканування `ntmap` дозволяє визначити, які служби доступні на сервері. Очікувано відкритими мають бути порти, потрібні для роботи SSH, Samba та NFS. Журналювання SSH перевіряється шляхом спроби входу під неіснуючим користувачем `fakeuser` і подальшим переглядом журналу через `journalctl`.

Десятим етапом є перевірка відмовостійкості. У межах лабораторного стенду вона полягає в перезавантаженні FileServer, UbuntuClient і Windows VM з подальшою перевіркою доступності служб і спільного ресурсу. Після перезавантаження FileServer повинні автоматично запускатися служби SSH, Samba та NFS. Після перезавантаження UbuntuClient має зберігатися IP-адреса Host-only інтерфейсу та автоматично монтуватися NFS-ресурс через `/etc/fstab`. Windows VM повинна знову мати можливість підключитися до SMB-ресурсу.

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		54

Таблиця 2.5

Методика тестування файлового сервера змішаного доступу

№	Група тестування	Інструмент / команда	Очікуваний результат
1	Перевірка мережі	ping, ip -br a	Вузли доступні в мережі 192.168.56.0/24
2	Перевірка служб	systemctl status ssh/smbd/nfs-server	Служби активні та запускаються після reboot
3	SMB-доступ	\192.168.56.10\mixed, net use	Windows-клієнт відкриває спільний ресурс
4	NFS-доступ	mount, mount grep mixed	UbuntuClient монтує /mnt/mixed
5	Змішаний доступ	створення файлів з Windows і Linux	Файли видно з обох клієнтів
6	Перевірка ivan	touch, echo, створення файлу в Windows	Користувач має читання і запис
7	Перевірка petro	touch, створення файлу в Windows	Користувач отримує Permission denied / Access is denied
8	SMB-блокування	smbstatus --locks	Відображаються відкриті файли або SMB lease
9	Продуктивність мережі	iperf3	Отримано швидкість передавання даних
10	Продуктивність NFS	dd	Отримано швидкість читання і запису через NFS
11	Продуктивність SMB	robocopy	Отримано швидкість читання і запису через SMB
12	Безпека портів	nmap -sV 192.168.56.10	Відкриті лише необхідні серверні служби
13	Журналювання SSH	journalctl -u ssh	Фіксуються невдалі спроби входу
14	NFS root_squash	sudo rm / touch на NFS-ресурсі	root клієнта не отримує надмірних прав
15	Відмовостійкість	reboot, повторна перевірка служб	Після перезавантаження система відновлює роботу

Запропонована методика тестування дозволяє комплексно оцінити працездатність файлового сервера змішаного доступу. Вона охоплює всі ключові аспекти: мережу, служби, SMB, NFS, права користувачів, змішаний доступ, блокування, продуктивність, безпеку та відновлення після перезавантаження. Саме ця методика використовується в третьому розділі під час практичної розробки та тестування системи.

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		55

Висновки до розділу 2

У другому розділі було виконано моделювання файлового сервера змішаного доступу SMB/NFS на базі Linux. На основі результатів аналізу предметної області було визначено основні компоненти системи, обґрунтовано вибір операційної системи, файлової системи, структури дискового простору, архітектури стенду, моделі прав доступу та методики тестування.

Як серверну операційну систему обрано Ubuntu Server. Такий вибір обґрунтовано стабільністю дистрибутива, широкою підтримкою програмних пакетів, зручністю адміністрування, наявністю документації та сумісністю з необхідними компонентами: Samba, NFS Server, OpenSSH, POSIX ACL та стандартними Linux-утилітами керування користувачами й правами доступу.

Для організації файлового простору обрано файлову систему ext4. Вона є стабільною, простою в адмініструванні, підтримує POSIX-права та POSIX ACL, а також повністю відповідає вимогам лабораторного стенду. У ролі центрального спільного ресурсу визначено каталог /srv/storage/mixed, який у подальшій практичній реалізації публікується через Samba та експортується через NFS.

Було спроектовано архітектуру лабораторного стенду, що складається з трьох віртуальних машин: FileServer, UbuntuClient та Windows VM. FileServer виконує роль центрального файлового сервера з IP-адресою 192.168.56.10, UbuntuClient використовується для перевірки NFS-доступу з адресою 192.168.56.20, а Windows VM — для перевірки SMB-доступу з адресою 192.168.56.30. Взаємодія вузлів здійснюється через ізольовану Host-only мережу 192.168.56.0/24.

Розроблено модель прав доступу та автентифікації. У межах лабораторного стенду використовується локальна модель на базі Linux-користувачів, Samba-облікових записів, узгоджених UID/GID і POSIX ACL. Користувач ivan має права читання та запису, користувач retro — лише читання, а адміністратор має повний доступ до налаштування системи. Для групового керування доступом передбачено групи fs_rw, fs_ro, fs_admins та fs_data.

Розглянуто можливість інтеграції з LDAP, Active Directory або FreeIPA. Встановлено, що для лабораторного стенду така інтеграція не є обов'язковою,

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		56

оскільки локальна модель користувачів дозволяє наочно продемонструвати принципи роботи SMB, NFS і POSIX ACL. Водночас для корпоративного середовища централізована автентифікація може бути доцільним напрямом подальшого розвитку системи.

Також було описано основні сценарії використання файлового сервера: підключення Windows-клієнта через SMB, підключення Linux-клієнта через NFS, створення файлів користувачем із правами запису, спроба запису користувачем лише для читання, перевірка видимості файлів між SMB і NFS, перевірка SMB-блокувань і перевірка автоматичного NFS-монтування після перезавантаження.

У завершальній частині розділу сформовано методику тестування системи. Вона включає перевірку мережевої доступності, стану служб, функціонального доступу через SMB і NFS, прав користувачів, змішаного доступу, блокувань, продуктивності, безпеки та відмовостійкості. Ця методика є основою для практичної реалізації та тестування системи в третьому розділі.

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		57

РОЗДІЛ 3 РОЗРОБКА ТА ТЕСТУВАННЯ СИСТЕМИ

3.1 Розгортання серверної ОС, підготовка дискової підсистеми та налаштування служби Samba

Практична реалізація системи почалася з розгортання серверної операційної системи. Як серверну платформу було використано Ubuntu Server, оскільки вона є стабільною, підтримує необхідні мережеві служби, має зручну систему керування пакетами та добре підходить для розгортання файлових серверів на базі Samba і NFS.

Після встановлення операційної системи було виконано базове налаштування мережі. FileServer отримав статичну IP-адресу 192.168.56.10 у Host-only мережі. Ця адреса використовується клієнтськими машинами для підключення до файлових ресурсів через SMB та NFS. Також сервер має NAT-інтерфейс для доступу до інтернету та встановлення необхідних пакетів.

Перевірка мережевих інтерфейсів FileServer здійснювалася за допомогою команди `ip -br a`. У результаті було підтверджено, що Host-only інтерфейс сервера активний і має IP-адресу 192.168.56.10.

```
ast login: Sat Jun 20 14:32:18 2026 from 192.168.56.1
administrator@fileserver:~$ ip -br a
lo                UNKNOWN                127.0.0.1/8 ::1/128
enp0s3            UP                    10.0.2.15/24 fd17:625c:f037:2:a00:27ff:fe6b:3b1d/64
enp0s8            UP                    192.168.56.10/24 fe80::a00:27ff:fe12:1d99/64
administrator@fileserver:~$ systemctl status ssh --no-pager
ssh.service - OpenBSD Secure Shell server
Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enabled)
Active: active (running) since Sat 2026-06-20 16:40:27 UTC; 18min ago
Invocation: 7c0062f249ec4ad18738a42cdf7dffd5
TriggeredBy: ● ssh.socket
Docs: man:sshd(8)
      man:sshd_config(5)
Process: 1577 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
Main PID: 1581 (sshd)
Tasks: 1 (limit: 1713)
Memory: 5.3M (peak: 7.1M)
CPU: 52ms
CGroup: /system.slice/ssh.service
        └─1581 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Jun 20 16:40:27 fileserver systemd[1]: Starting ssh.service - OpenBSD Secure Shell server:
Jun 20 16:40:27 fileserver sshd[1581]: Server listening on 0.0.0.0 port 22.
Jun 20 16:40:27 fileserver systemd[1]: Started ssh.service - OpenBSD Secure Shell server:
Jun 20 16:40:27 fileserver sshd[1581]: Server listening on :: port 22.
Jun 20 16:58:11 fileserver sshd-session[1839]: Accepted password for administrator:
Jun 20 16:58:11 fileserver sshd-session[1839]: pam_unix(sshd:session): session opened for
Administrator: Some lines were ellipsized, use -l to show in full.
administrator@fileserver:~$
```

Рисунок 3.1 — Мережеві інтерфейси FileServer після налаштування IP-адреси

На наступному етапі було підготовлено дискову структуру файлового сервера. Для розміщення спільного ресурсу використано каталог /srv/storage/mixed. Каталог /srv у Linux традиційно застосовується для даних сервісів, які надаються системою. Тому розміщення файлового ресурсу саме в /srv/storage є логічним і зручним для адміністрування.

Структура каталогів була організована таким чином:

```
/srv/storage
/srv/storage/mixed
/srv/storage/mixed/archive
/srv/storage/mixed/projects
/srv/storage/mixed/public
```

Каталог /srv/storage/mixed є основним спільним каталогом, який у подальшому публікується через Samba для Windows-клієнтів і експортується через NFS для Linux-клієнтів. Додаткові підкаталоги archive, projects і public були створені для демонстрації організації файлового простору та перевірки доступу з різних клієнтських систем.

```
administrator@fileserver:~$ ls -la /srv/storage/mixed
total 327744
drwxrws---+6 root fs_data      4096 Jun 20 18:01 .
drwxr-xr-x  3 root root        4096 Jun 20 14:17 ..
drwxrws---+2 root fs_data      4096 Jun 20 14:17 archive
-rw-rw----+1 ivan fs_data       85 Jun 20 18:16 concurrent_test.txt
-rw-rw----+1 ivan fs_data 67108864 Jun 20 17:07 dd_nfs_test.dat
-rw-rw----+1 ivan fs_data       0 Jun 20 15:18 nfs_ivan_test.txt
drwxrws---+2 root fs_data      4096 Jun 20 14:17 projects
drwxrws---+2 root fs_data      4096 Jun 20 14:17 public
-rwxrwx---+1 ivan fs_data 268435456 Jun 20 17:45 smb_test_256mb.bin
drwxrws---+2 ivan fs_data      4096 Jun 20 14:29 test_from_smb
-rwxrwx---+1 ivan fs_data       0 Jun 20 15:44 windows_smb_test.txt
administrator@fileserver:~$
```

Рисунок 3.2 — Структура спільного каталогу /srv/storage/mixed на FileServer

Для реалізації розмежування прав доступу було створено локальних користувачів і групи. Основними користувачами для тестування стали ivan і petro. Користувач ivan призначений для перевірки доступу з правами читання та запису, а користувач petro — для перевірки доступу лише на читання. Така модель дозволяє перевірити, чи правильно працює обмеження прав незалежно від того, через який протокол користувач підключається до спільного каталогу.

Для групового керування доступом було створено групи fs_rw, fs_ro, fs_admins та fs_data. Група fs_rw використовується для користувачів із правом

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		59

запису, `fs_ro` — для користувачів із правом лише читання, `fs_admins` — для адміністраторів ресурсу, а `fs_data` — як загальна група файлового ресурсу. Використання груп спрощує адміністрування, оскільки права задаються не кожному користувачеві окремо, а групам.

Після створення користувачів і груп було налаштовано POSIX ACL для каталогу `/srv/storage/mixed`. Це дозволило надати різні права різним групам: група `fs_rw` отримала права читання, запису та виконання; група `fs_ro` — права читання та виконання; група `fs_admins` — повні права. Також було налаштовано default ACL, щоб нові файли й каталоги автоматично успадковували необхідні права доступу.

```
administrator@fileserv:~$ getfacl /srv/storage/mixed
getfacl: Removing leading '/' from absolute path names
# file: srv/storage/mixed
# owner: root
# group: fs_data
# flags: -s-
user::rwx
group::r-x
group:fs_rw:rwx
group:fs_ro:r-x
group:fs_admins:rwx
mask::rwx
other::---
default:user::rwx
default:group::r-x
default:group:fs_rw:rwx
default:group:fs_ro:r-x
default:group:fs_admins:rwx
default:mask::rwx
default:other::---
administrator@fileserv:~$
```

Рисунок 3.3 — POSIX ACL спільного каталогу `/srv/storage/mixed`

Після підготовки файлової структури було встановлено та налаштовано Samba. Samba є ключовим компонентом системи для забезпечення доступу Windows-клієнтів до Linux-файлового сервера. Для встановлення необхідних пакетів на FileServer використовувалися стандартні засоби керування пакетами Ubuntu.

Основні команди, що використовувалися для встановлення Samba:

sudo apt update

sudo apt install -y samba acl

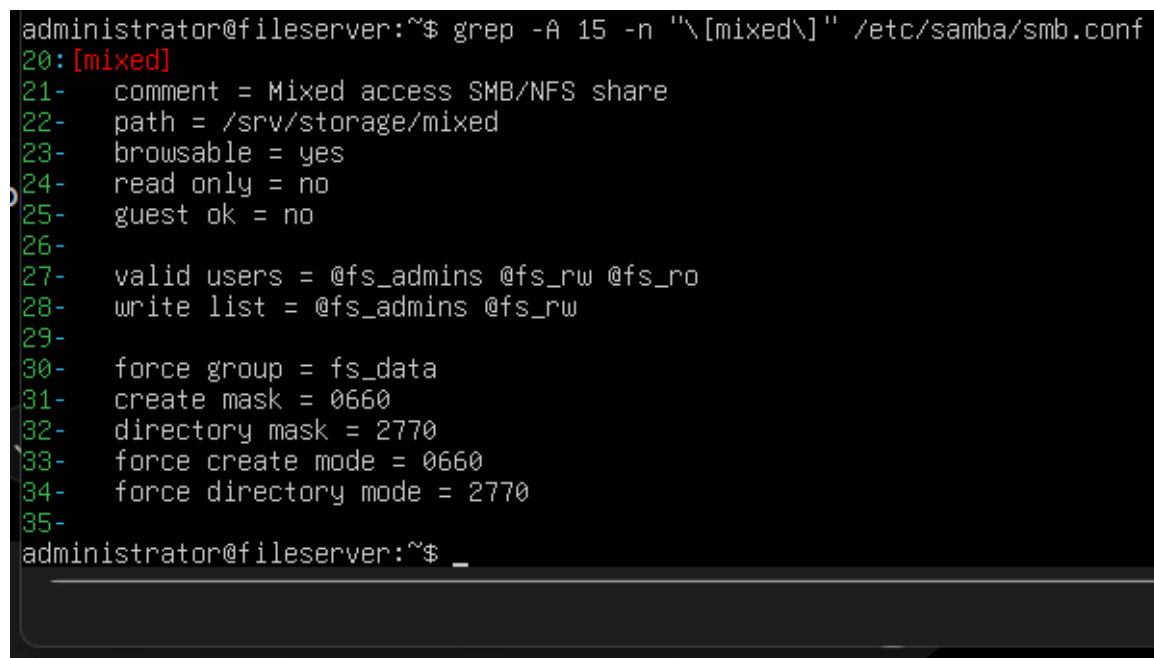
		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				60
Змн.	Арк.	№ докум.	Підпис	Дата		

Після встановлення служби було виконано налаштування конфігураційного файлу `/etc/samba/smb.conf`. У цьому файлі було створено окремий SMB-ресурс `mixed`, який вказує на каталог `/srv/storage/mixed`. Саме цей ресурс відкривається Windows-клієнтом за адресою `\192.168.56.10\mixed`.

Приклад логіки конфігурації ресурсу `mixed`:

```
[mixed]
path = /srv/storage/mixed
browseable = yes
read only = no
valid users = ivan petro
create mask = 0660
directory mask = 0770
```

Параметр `path` визначає шлях до серверного каталогу, який буде доступний через SMB. Параметр `browseable` дозволяє відображати ресурс у мережевому оточенні. Параметр `read only = no` дозволяє запис для користувачів, яким це дозволено правами файлової системи. Параметр `valid users` обмежує доступ лише визначеними користувачами. Параметри `create mask` і `directory mask` визначають базові права для нових файлів і каталогів, створених через SMB.



```
administrator@fileservеr:~$ grep -A 15 -n "[mixed\\]" /etc/samba/smb.conf
20: [mixed]
21-  comment = Mixed access SMB/NFS share
22-  path = /srv/storage/mixed
23-  browsable = yes
24-  read only = no
25-  guest ok = no
26-
27-  valid users = @fs_admins @fs_rw @fs_ro
28-  write list = @fs_admins @fs_rw
29-
30-  force group = fs_data
31-  create mask = 0660
32-  directory mask = 2770
33-  force create mode = 0660
34-  force directory mode = 2770
35-
administrator@fileservеr:~$ _
```

Рисунок 3.4 — Фрагмент конфігурації Samba-ресурсу `mixed` у файлі `/etc/samba/smb.conf`

Важливо зазначити, що параметри Samba не скасовують прав Linux-файлової системи. Навіть якщо в smb.conf ресурс не є read only, користувач усе одно не зможе записати файл, якщо POSIX ACL на серверному каталозі не дозволяє йому запис. Саме тому в системі змішаного доступу Samba-конфігурація та POSIX ACL повинні бути узгоджені між собою.

Для доступу через SMB користувачі ivan і petro були додані до бази Samba. Це потрібно тому, що Windows-клієнт автентифікується не лише як Linux-користувач, а і як Samba-користувач. Для цього використовувалася команда smbpasswd. Після додавання користувачів Samba може приймати підключення з Windows VM під відповідними обліковими записами.

Після внесення змін у конфігурацію Samba було перезапущено службу smbd. Перевірка стану служби виконувалася за допомогою systemctl. Активний стан smbd підтверджує, що Samba запущена та готова приймати підключення від Windows-клієнтів.

```

administrator@fileserv:~$ systemctl status smbd --no-pager
● smbd.service - Samba SMB Daemon
   Loaded: loaded (/usr/lib/systemd/system/smbd.service; enabled; preset: enabled)
   Active: active (running) since Sun 2026-06-21 00:23:45 UTC; 15min ago
  Invocation: f8087a9053914fc58546f4ddb859720
    Docs: man:smbd(8)
          man:samba(7)
          man:smb.conf(5)
  Process: 1551 ExecCondition=/usr/share/samba/is-configured smb (code=exited, status=0/SUCCESS)
  Process: 1589 ExecStartPre=/usr/share/samba/update-apparmor-samba-profile (code=exited, status=0/SUCCESS)
 Main PID: 1594 (smbd)
   Status: "smbd: ready to serve connections..."
    Tasks: 3 (limit: 1713)
  Memory: 14.4M (peak: 15.1M)
    CPU: 112ms
   CGroup: /system.slice/smbd.service
           └─1594 /usr/sbin/smbd --foreground --no-process-group
             └─1604 "smbd: notifyd"
               └─1606 "smbd: cleanupd"

чер 21 00:23:44 fileserv systemd[1]: Starting smbd.service - Samba SMB Daemon...
чер 21 00:23:45 fileserv systemd[1]: Started smbd.service - Samba SMB Daemon.
administrator@fileserv:~$

```

Рисунок 3.5 — Стан служби smbd після налаштування Samba

Таким чином, у межах підрозділу було виконано розгортання серверної ОС, налаштування мережевої адресації, підготовку каталогу /srv/storage/mixed, створення користувачів і груп, налаштування POSIX ACL та конфігурування Samba-ресурсу mixed. Це створило основу для подальшого налаштування NFS і перевірки змішаного доступу з Windows- та Linux-клієнтів.

3.2 Налаштування NFS-сервера та механізмів спільного змішаного доступу і блокувань

Після підготовки серверної файлової структури та налаштування служби Samba наступним етапом стало налаштування NFS-сервера. NFS використовується для забезпечення доступу Linux-клієнта UbuntuClient до того самого спільного каталогу /srv/storage/mixed, який уже був опублікований для Windows-клієнта через SMB. Таким чином, один фізичний каталог на FileServer стає доступним через два різні протоколи: SMB/CIFS для Windows VM та NFS для UbuntuClient.

Для роботи NFS на FileServer було встановлено пакет nfs-kernel-server. Він забезпечує експорт серверних каталогів у мережу та дозволяє клієнтським Linux-системам монтувати ці каталоги у власну локальну файлову систему. У межах лабораторного стенду NFS-ресурс було надано для Host-only мережі 192.168.56.0/24, у якій знаходяться FileServer, UbuntuClient і Windows VM.

Основним конфігураційним файлом NFS-сервера є /etc/exports. Саме в ньому визначається, які каталоги експортуються, для яких клієнтів вони доступні та з якими параметрами. Для спільного каталогу /srv/storage/mixed було задано експорт для мережі 192.168.56.0/24. Це означає, що всі клієнти в межах цієї мережі можуть звертатися до NFS-ресурсу відповідно до налаштованих параметрів доступу.

Фрагмент налаштування NFS-експорту має такий вигляд:

```
/srv/storage/mixed
192.168.56.0/24(rw,sync,no_subtree_check,root_squash,sec=sys)
```

Параметр rw дозволяє операції читання та запису для тих користувачів, яким це дозволено правами файлової системи. Параметр sync забезпечує синхронний запис даних, що підвищує надійність, оскільки сервер підтверджує операції після фактичного запису. Параметр no_subtree_check вимикає додаткову перевірку підкаталогів, що спрощує роботу з експортованим каталогом і зменшує ризик помилок під час доступу. Параметр root_squash є важливим механізмом безпеки, який не дозволяє користувачу root на клієнтській машині отримати root-права на серверному ресурсі. Параметр sec=sys вказує на використання стандартної Unix-моделі автентифікації.

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		63

```

administrator@fileserver:~$ sudo exportfs -v
[sudo: authenticate] Password:
/srv/storage/mixed
192.168.56.0/24(sync,wdelay,hide,no_subtree_check,sec=sys,rw,secure,root_squash,no_all_squash)
administrator@fileserver:~$ _

```

Рисунок 3.6 — Конфігурація NFS-експорту спільного каталогу /srv/storage/mixed

Після внесення змін до /etc/exports було застосовано налаштування NFS за допомогою команди `exportfs -ra`. Для перевірки активних експортів використовувалася команда `exportfs -v`, яка показує фактично опубліковані NFS-ресурси та їх параметри. У результаті було підтверджено, що каталог /srv/storage/mixed експортується для мережі 192.168.56.0/24.

Окремо було перевірено стан служби `nfs-server`. Активний стан служби підтверджує, що сервер готовий приймати NFS-запити від Linux-клієнтів. У Linux статус `nfs-server` може відображатися як `active (exited)`, що є нормальним для цієї служби, оскільки частина роботи виконується допоміжними процесами ядра та службами RPC.

```

administrator@fileserver:~$ systemctl status nfs-server --no-pager
• nfs-server.service - NFS server and services
   Loaded: loaded (/usr/lib/systemd/system/nfs-server.service; enabled; preset: enabled)
   Drop-In: /run/systemd/generator/nfs-server.service.d
            └─order-with-mounts.conf
   Active: active (exited) since Sun 2026-06-21 00:23:44 UTC; 36min ago
   Invocation: 3e8737f4c22e494d954946c85eb3d587
      Docs: man:rpc.nfsd(8)
            man:exportfs(8)
   Process: 1357 ExecStartPre=/usr/sbin/exportfs -r (code=exited, status=0/SUCCESS)
   Process: 1379 ExecStart=/bin/sh -c /usr/sbin/nfsdctl autostart || /usr/sbin/rpc.nfsd (code=exited, status=0/SUCCESS)
   Main PID: 1379 (code=exited, status=0/SUCCESS)
  Mem peak: 1.8M
    CPU: 28ms

чеп 21 00:23:43 fileserver systemd[1]: Starting nfs-server.service - NFS server and services...
чеп 21 00:23:44 fileserver systemd[1]: Finished nfs-server.service - NFS server and services.
administrator@fileserver:~$

```

Рисунок 3.7 — Стан служби `nfs-server` після налаштування NFS

На клієнтській машині `UbuntuClient` було створено точку монтування /mnt/mixed. Саме в цей каталог підключається віддалений ресурс 192.168.56.10:/srv/storage/mixed. Після монтування користувачі Linux-клієнта можуть працювати з файлами у /mnt/mixed так, ніби вони знаходяться в локальній файловій системі. Насправді ж усі операції виконуються з файлами, які фізично зберігаються на FileServer.

Для ручного монтування ресурсу може використовуватися команда:

`sudo mount -t nfs 192.168.56.10:/srv/storage/mixed /mnt/mixed`

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		64

У практичній реалізації також було налаштовано автоматичне монтування NFS-ресурсу після перезавантаження UbuntuClient. Для цього в файл `/etc/fstab` було додано запис, який визначає серверний ресурс, локальну точку монтування, тип файлової системи та параметри монтування.

Приклад запису у `/etc/fstab`:

```
192.168.56.10:/srv/storage/mixed /mnt/mixed nfs defaults,_netdev 0 0
```

Параметр `_netdev` вказує, що ресурс є мережевим і повинен монтуватися після ініціалізації мережі. Це важливо, оскільки без активного мережевого інтерфейсу NFS-ресурс не може бути підключений. Після внесення змін до `/etc/fstab` було виконано команду `systemctl daemon-reload`, а потім `mount -a` для перевірки правильності запису.

```

administrator@administrator-VirtualBox: ~
administrator@administrator-VirtualBox:~$ cat /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options>          <dump> <pass>
# / was on /dev/sda2 during curtin installation
/dev/disk/by-uuid/a633ad9f-2136-45e0-ae7d-7bb001e3d739 / ext4 defaults 0 1
/swap.img      none      swap      sw        0          0
192.168.56.10:/srv/storage/mixed /mnt/mixed nfs defaults,_netdev 0 0
administrator@administrator-VirtualBox:~$
  
```

Рисунок 3.8 — Запис автоматичного монтування NFS-ресурсу у файлі `/etc/fstab` на UbuntuClient

Факт успішного монтування перевірявся за допомогою команди `mount | grep mixed`. У результаті було підтверджено, що ресурс `192.168.56.10:/srv/storage/mixed` змонтований у каталог `/mnt/mixed` з використанням NFS. Додатково команда `ls -la /mnt/mixed` показала вміст спільного каталогу, зокрема підкаталоги `archive`, `projects`, `public` та тестові файли, створені з різних клієнтів.

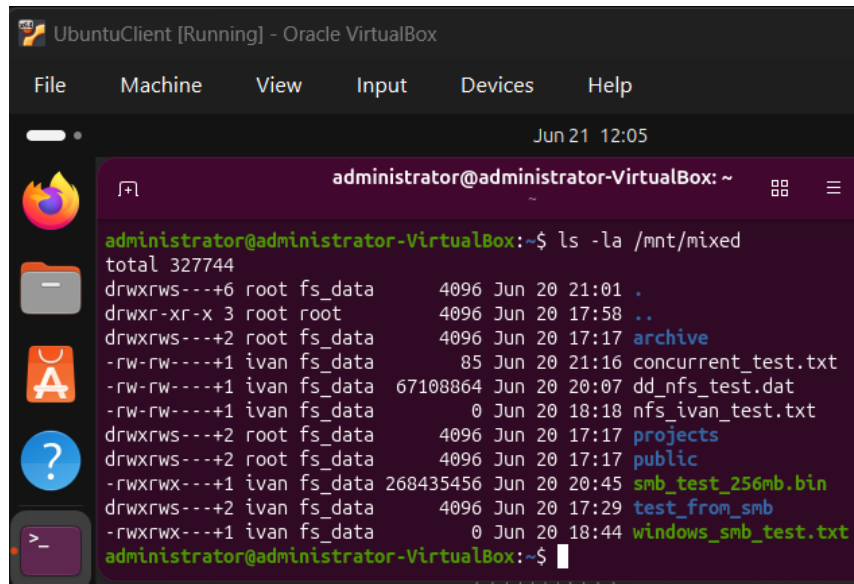


Рисунок 3.9 — Активне NFS-монтування спільного каталогу на UbuntuClient

Особливістю реалізованої системи є те, що Samba та NFS працюють з одним і тим самим серверним каталогом /srv/storage/mixed. Це створює змішаний доступ: Windows-клієнт відкриває ресурс через SMB, а Linux-клієнт монтує той самий каталог через NFS. Завдяки цьому файл, створений через Windows VM, стає видимим на UbuntuClient, а файл, створений через UbuntuClient, відображається у Windows VM. Така поведінка підтверджує, що обидва протоколи використовують єдину файлову область, а не окремі копії даних.

Разом із тим змішаний доступ потребує врахування блокувань і кешування. SMB і NFS мають різні механізми роботи з відкритими файлами. Samba підтримує SMB-блокування, oplocks і leases, які дозволяють відстежувати відкриті файли з боку Windows-клієнтів. NFS використовує іншу модель доступу, орієнтовану на Unix/Linux-системи. Тому для коректної роботи важливо не лише надати доступ, а й перевірити, як система поводить себе під час відкриття файлів і одночасної роботи з ними.

Для аналізу SMB-блокувань на FileServer використовувалася команда smbstatus --locks. Вона показує, які файли відкриті через SMB, який користувач їх відкрив, який режим доступу використовується та чи застосовується механізм lease. У межах тестування було відкрито файл зі спільного ресурсу у Windows VM, після чого на FileServer виконано перевірку блокувань. У результаті було зафіксовано

активний SMB lease для користувача з ідентифікатором 1101, що відповідає користувачу ivan.

```

administrator@fileserv:~$ sudo smbstatus --locks
Locked files:
Pid      User(ID)  DenyMode Access      R/W      Oplock      SharePath  Name  Time
-----
2020     1101     DENY_NONE 0x100081    RONLY    LEASE()     /srv/storage/mixed .   Sat Jun 20 18:13:45 2026
2020     1101     DENY_NONE 0x100081    RONLY    LEASE()     /srv/storage/mixed .   Sat Jun 20 18:13:45 2026
administrator@fileserv:~$

```

Рисунок 3.10 — Перевірка SMB-блокувань за допомогою smbstatus --locks

Наявність записів у smbstatus --locks підтверджує, що Samba відстежує стан відкритих файлів з боку SMB-клієнтів. У виводі команди відображаються такі поля, як PID процесу, ідентифікатор користувача, режим доступу, тип операції читання або запису та стан lease/oplock. Це дозволяє адміністратору контролювати активність SMB-клієнтів і діагностувати можливі проблеми з доступом до файлів.

Водночас слід зазначити, що SMB-блокування не є повною заміною загальної системи контролю одночасного доступу між SMB і NFS. Воно насамперед відображає стан SMB-клієнтів. NFS-клієнти можуть працювати з тим самим файлом через інший механізм доступу. Тому в системах змішаного доступу важливо правильно налаштовувати права, обмежувати небажаний запис, застосовувати POSIX ACL і визначати правила роботи користувачів із критичними файлами.

Додатковим механізмом безпеки NFS є root_squash. Його робота полягає в тому, що користувач root на клієнтській машині не отримує автоматично root-права на серверному файловому ресурсі. У практичній реалізації це проявилось під час спроби виконати операції з файлами через sudo на UbuntuClient. Якщо користувач не має відповідних прав на сервері, операція завершується повідомленням Permission denied. Це підтверджує, що NFS не надає клієнтському root надмірних прав на сервері.

Таким чином, у межах цього підрозділу було налаштовано NFS-сервер, експортовано каталог /srv/storage/mixed для мережі 192.168.56.0/24, налаштовано автоматичне монтування ресурсу на UbuntuClient, перевірено активне NFS-підключення та проаналізовано механізми змішаного доступу і SMB-блокувань. Це

забезпечило повноцінну основу для подальшого функціонального тестування доступу з Windows- та Linux-клієнтів.

3.3 Функціональне тестування (доступ із Windows- та Linux-клієнтів, перевірка прав, одночасний запис)

Після налаштування Samba та NFS було виконано функціональне тестування файлового сервера змішаного доступу. Метою цього етапу є перевірка практичної працездатності системи, підтвердження доступу до спільного каталогу з Windows- та Linux-клієнтів, перевірка прав користувачів, а також підтвердження того, що SMB і NFS працюють з одним і тим самим серверним каталогом /srv/storage/mixed.

Функціональне тестування виконувалося у віртуалізованому середовищі, яке включає три вузли: FileServer, UbuntuClient та Windows VM. FileServer має IP-адресу 192.168.56.10 і виконує роль центрального файлового сервера. Windows VM використовується для перевірки SMB-доступу, а UbuntuClient — для перевірки NFS-доступу. Такий підхід дозволяє змодельовати гетерогенну мережу, у якій різні клієнтські операційні системи працюють із єдиним файловим ресурсом.

Першим етапом функціонального тестування стала перевірка доступу з Windows-клієнта. Для цього у Windows VM було відкрито мережевий шлях \192.168.56.10\mixed. Після введення облікових даних користувача ivan Windows-клієнт отримав доступ до спільного каталогу. У Провіднику Windows відобразилися підкаталоги archive, projects, public, а також тестові файли, створені в межах перевірки. Це підтверджує, що служба Samba працює коректно, а SMB-ресурс mixed доступний для Windows-клієнтів.

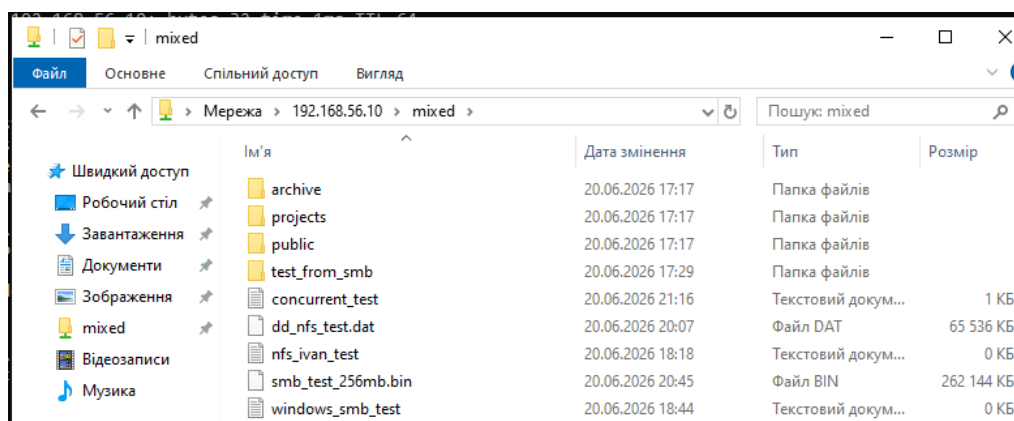


Рисунок 3.11 — Доступ Windows-клієнта до SMB-ресурсу \192.168.56.10\mixed

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				68
Змн.	Арк.	№ докум.	Підпис	Дата		

Додатково SMB-ресурс було підключено як мережевий диск за допомогою команди `net use`. Такий спосіб підключення зручний для тестування, оскільки дозволяє працювати зі спільним ресурсом як із локальним диском Windows. У межах роботи ресурс було підключено до диска Z: під користувачем `ivan`. Успішне виконання команди `net use` підтверджує, що Windows-клієнт може автентифікуватися на Samba-сервері та отримувати доступ до ресурсу.

```

Win10Client [Running] - Oracle VirtualBox
File Machine View Input Devices Help

Administrator: Командний рядок

C:\Windows\system32>net use Z: /delete
The network connection could not be found.

More help is available by typing NET HELPMSG 2250.

C:\Windows\system32>net use Z: \\192.168.56.10\mixed /user:ivan
The command completed successfully.

C:\Windows\system32>
  
```

Рисунок 3.12 — Підключення SMB-ресурсу `mixed` як мережевого диска у Windows VM

Наступним етапом було перевірено доступ із Linux-клієнта `UbuntuClient` через NFS. На клієнтській машині віддалений ресурс `192.168.56.10:/srv/storage/mixed` був змонтований у каталог `/mnt/mixed`. Перевірка командою `mount | grep mixed` показала, що ресурс активний і підключений як NFS-файлова система. Команда `ls -la /mnt/mixed` відобразила вміст спільного каталогу, що підтверджує доступність ресурсу для Linux-клієнта.

```

UbuntuClient [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Jun 21 12:04
administrator@administrator-VirtualBox: ~
administrator@administrator-VirtualBox:~$ sudo mount -a
[sudo: authenticate] Password:
administrator@administrator-VirtualBox:~$ mount | grep mixed
192.168.56.10:/srv/storage/mixed on /mnt/mixed type nfs4 (rw,relatime,vers=4.2,rsize
=524288,wsz=524288,namlen=255,hard,fatal_nerrors=none,proto=tcp,timeo=600,retran
s=2,sec=sys,clientaddr=192.168.56.20,local_lock=none,addr=192.168.56.10,_netdev)
administrator@administrator-VirtualBox:~$ ls -la /mnt/mixed
total 327744
drwxrws---+6 root fs_data      4096 Jun 20 21:01 .
drwxr-xr-x  3 root root        4096 Jun 20 17:58 ..
drwxrws---+2 root fs_data      4096 Jun 20 17:17 archive
-rw-rw----+1 ivan fs_data       85 Jun 20 21:16 concurrent_test.txt
-rw-rw----+1 ivan fs_data    67108864 Jun 20 20:07 dd_nfs_test.dat
-rw-rw----+1 ivan fs_data       0 Jun 20 18:18 nfs_ivan_test.txt
drwxrws---+2 root fs_data      4096 Jun 20 17:17 projects
drwxrws---+2 root fs_data      4096 Jun 20 17:17 public
-rwxrwx---+1 ivan fs_data 268435456 Jun 20 20:45 smb_test_256mb.bin
drwxrws---+2 ivan fs_data      4096 Jun 20 17:29 test_from_smb
-rwxrwx---+1 ivan fs_data       0 Jun 20 18:44 windows_smb_test.txt
administrator@administrator-VirtualBox:~$

```

Рисунок 3.13 — Відображення спільного каталогу mixed на UbuntuClient через NFS

Для перевірки змішаного доступу було створено тестовий файл із боку Windows-клієнта. Після створення файлу через SMB він був перевірений на UbuntuClient у каталозі /mnt/mixed. Файл, створений у Windows, успішно відобразився на Linux-клієнті, що підтверджує використання одного спільного серверного каталогу, а не окремих копій даних для різних протоколів.

У зворотному напрямку було створено файл із боку UbuntuClient через NFS. Для цього користувач ivan виконав створення тестового файлу в каталозі /mnt/mixed. Після цього файл було перевірено у Windows VM через SMB-ресурс \192.168.56.10\mixed. Файл успішно відобразився у Провіднику Windows, що підтверджує двосторонню видимість даних між SMB- та NFS-клієнтами.

```

administrator@administrator-VirtualBox:~$ sudo -u ivan touch /mnt/mixed/nfs_from_ubuntu_visible_in_windows.txt
administrator@administrator-VirtualBox:~$

```

Рисунок 3.14 — Файл, створений через NFS на UbuntuClient

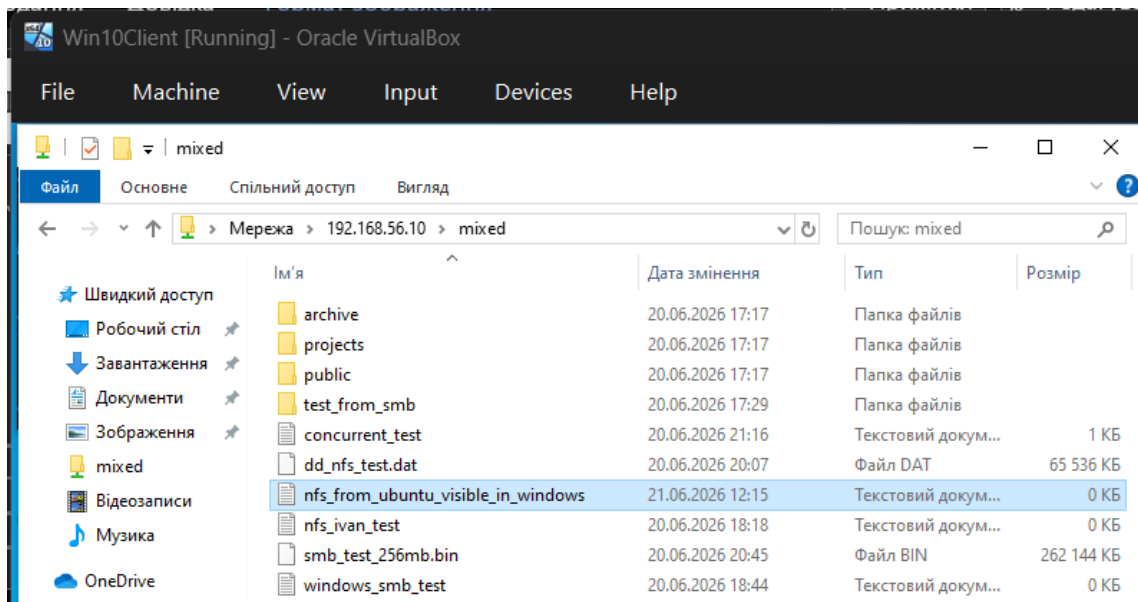


Рисунок 3.15 — Файл, створений через NFS на UbuntuClient, відображається у Windows-клієнті

Окремо було перевірено права доступу користувача *ivan*. Цей користувач входить до групи з правами читання і запису, тому він повинен мати можливість створювати файли у спільному каталозі. На UbuntuClient було виконано команду створення файлу від імені *ivan* у каталозі `/mnt/mixed`. Операція завершилася успішно, а файл з'явився у спільному каталозі. Це підтверджує, що права користувача *ivan* налаштовані коректно.

Приклад команди для перевірки запису користувача *ivan*:

`sudo -u ivan touch /mnt/mixed/nfs_ivan_test.txt`

Після виконання команди було перевірено вміст каталогу `/mnt/mixed`. Наявність файлу `nfs_ivan_test.txt` підтвердила успішний запис через NFS.

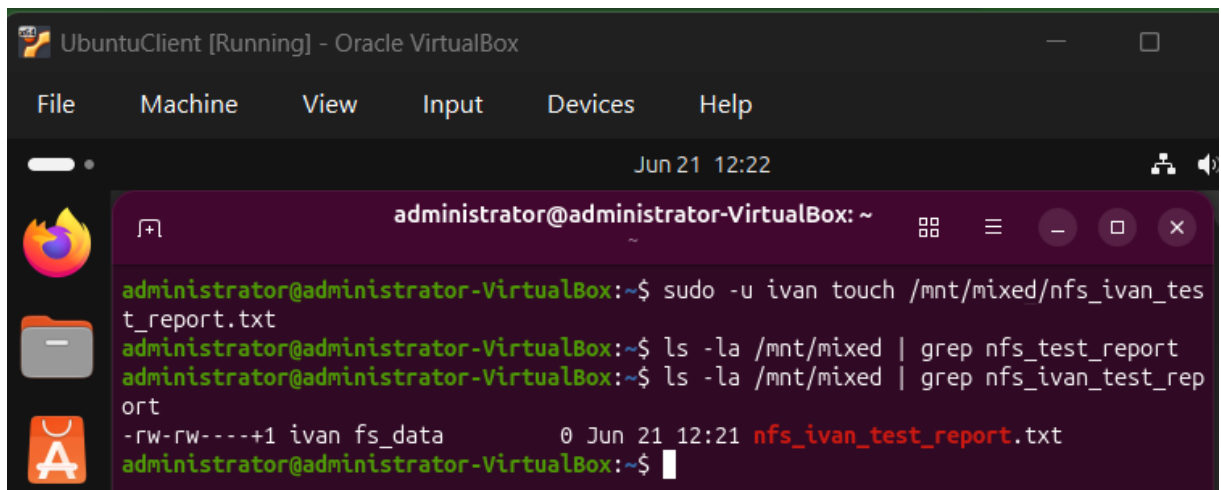


Рисунок 3.16 — Успішне створення файлу користувачем ivan через NFS

Після цього було перевірено права користувача petro. Відповідно до моделі доступу, користувач petro має лише право читання і не повинен мати можливості створювати або змінювати файли у спільному каталозі. На UbuntuClient було виконано спробу створення файлу від імені petro. У результаті система повернула повідомлення *Permission denied*, що підтверджує коректну роботу обмеження прав доступу через NFS.

Приклад команди для перевірки обмеження користувача petro:

sudo -u petro touch /mnt/mixed/petro_security_test.txt

Очікуваний результат:

touch: cannot touch '/mnt/mixed/petro_security_test.txt': Permission denied

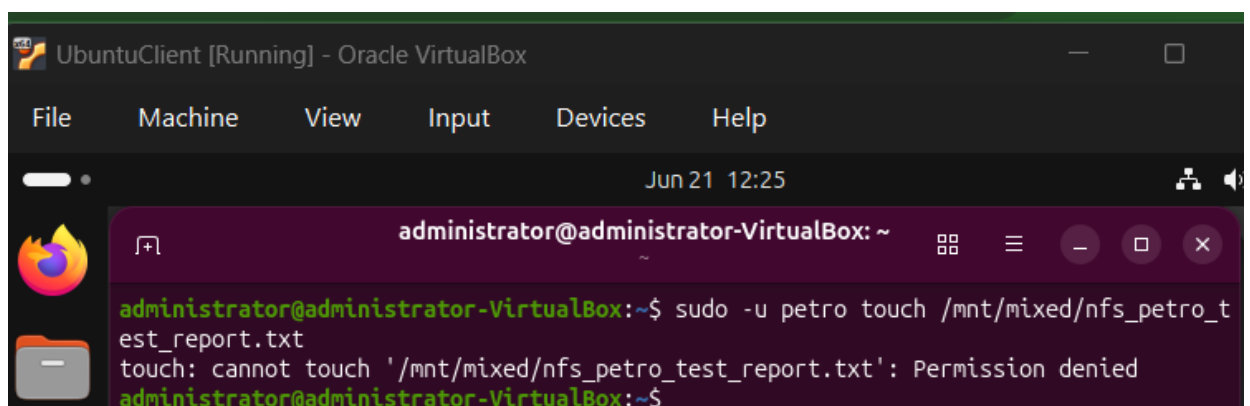


Рисунок 3.17 — Відмова у створенні файлу користувачем petro через NFS

Аналогічну перевірку було виконано з Windows-клієнта через SMB. Після підключення до ресурсу під користувачем petro було виконано спробу створити файл у мережевому каталозі. Система повернула повідомлення *Access is denied*. Це підтверджує, що обмеження прав застосовується не лише для NFS, а й для SMB-

доступу. Таким чином, POSIX ACL на серверному каталозі коректно контролюють доступ незалежно від протоколу підключення.

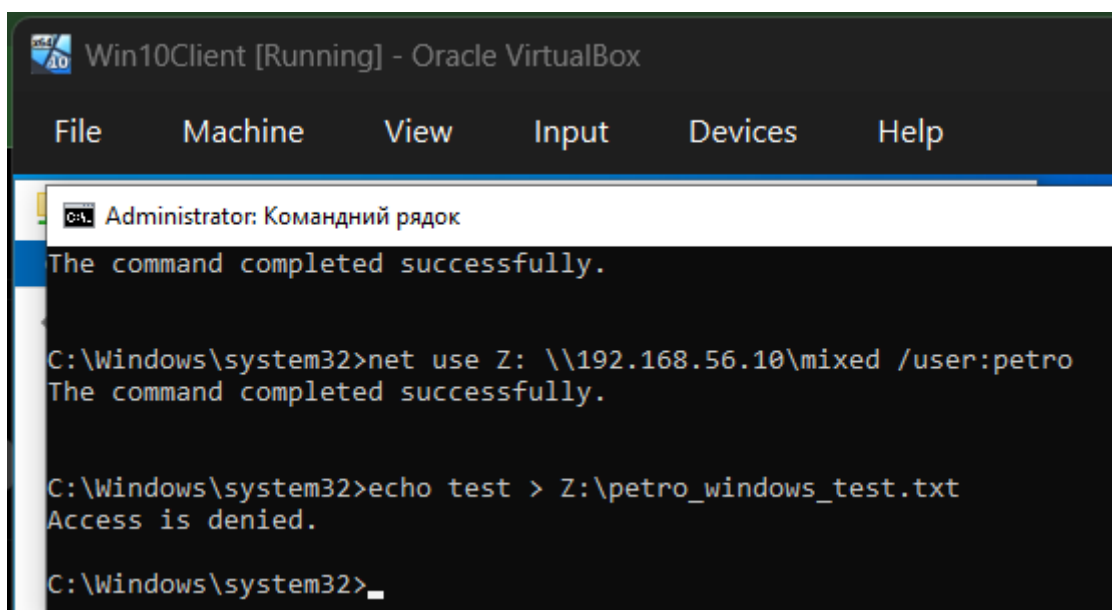


Рисунок 3.18 — Відмова у створенні файлу користувачем petro через SMB

Для перевірки одночасного змішаного доступу було створено тестовий файл `concurrent_test.txt`. Спочатку файл було створено або змінено з боку UbuntuClient через NFS, після чого він був відкритий у Windows VM через SMB. Далі у файл було додано рядок із боку Windows-клієнта, після чого зміни були перевірені на UbuntuClient за допомогою команди `cat`. Потім у зворотному напрямку було додано рядок із боку Linux-клієнта, а результат перевірено у Windows.

Приклад створення тестового файлу на UbuntuClient:

```
sudo -u ivan bash -c 'echo "Line 1 from NFS client" > /mnt/mixed/concurrent_test.txt'
```

Після редагування файлу у Windows через SMB на UbuntuClient виконується перевірка:

```
cat /mnt/mixed/concurrent_test.txt
```

Очікуваний результат полягає в тому, що рядки, додані через Windows-клієнт, стають видимими на Linux-клієнті. Це підтверджує працездатність змішаного доступу на рівні одного спільного каталогу.

```
administrator@administrator-VirtualBox:~$ sudo -u ivan bash -c 'echo "Line 1 from NFS client" > /mnt/mixed/concurrent_test.txt'
administrator@administrator-VirtualBox:~$ cat /mnt/mixed/concurrent_test.txt
Line 1 from NFS client
administrator@administrator-VirtualBox:~$
```

Рисунок 3.19 — Створення файлу concurrent_test.txt та запис першого рядка з NFS-клієнта

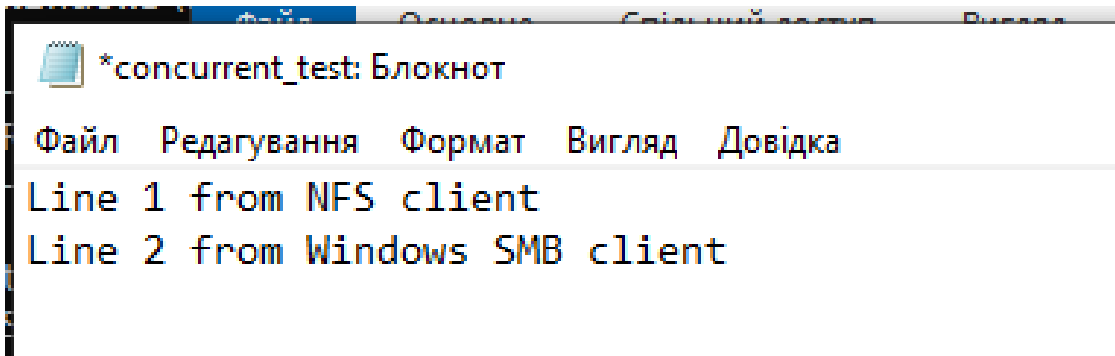


Рисунок 3.20 — Додавання другого рядка до файлу з боку Windows-клієнта через SMB

```
administrator@administrator-VirtualBox:~$ cat /mnt/mixed/concurrent_test.txt
Line 1 from NFS client
Line 2 from Windows SMB client
administrator@administrator-VirtualBox:~$
```

Рисунок 3.21 — Відображення внесених змін на NFS-клієнті командою cat

```
administrator@administrator-VirtualBox:~$ sudo -u ivan bash -c 'echo "Line 3 from NFS after SMB edit" >> /mnt/mixed/concurrent_test.txt'
administrator@administrator-VirtualBox:~$
```

Рисунок 3.22 — Додавання третього рядка до файлу з NFS-клієнта після редагування через SMB

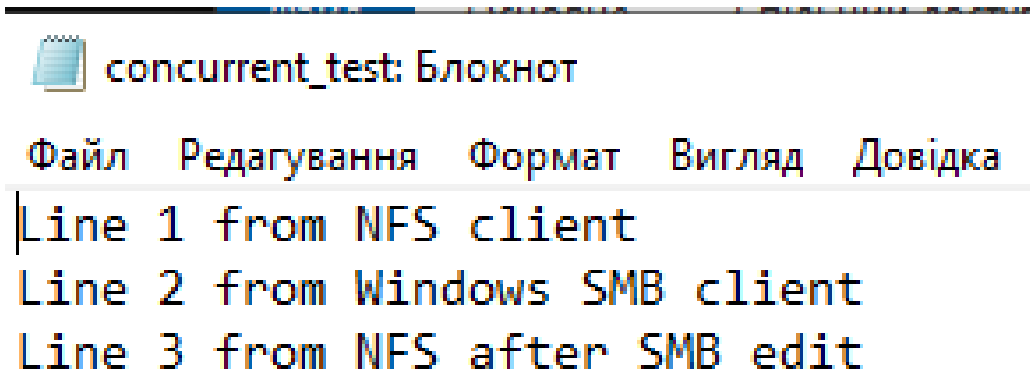


Рисунок 3.23 — Підсумковий вміст файлу concurrent_test.txt з усіма змінами у Windows-клієнті

Окремо слід зазначити, що одночасне редагування одного й того самого файлу через різні протоколи потребує обережності. SMB і NFS мають різні механізми кешування та блокування, тому в реальному середовищі для критичних документів доцільно уникати одночасного запису з різних клієнтів або використовувати прикладні засоби контролю версій. У межах лабораторного стенду тест підтвердив базову видимість змін між SMB і NFS.

Результати функціонального тестування узагальнено в таблиці 3.1.

Таблиця 3.1

Результати функціонального тестування файлового сервера

№	Тест	Клієнт	Протокол	Очікуваний результат	Фактичний результат
1	Відкриття ресурсу mixed у Windows	Windows VM	SMB	Ресурс відкривається	Успішно
2	Підключення ресурсу як диска Z:	Windows VM	SMB	Диск підключається	Успішно
3	Монтування ресурсу /mnt/mixed	UbuntuClient	NFS	Ресурс монтується	Успішно
4	Створення файлу користувачем ivan	UbuntuClient	NFS	Файл створюється	Успішно
5	Спроба запису користувачем petro	UbuntuClient	NFS	Відмова в доступі	Permission denied
6	Спроба запису користувачем petro	Windows VM	SMB	Відмова в доступі	Access is denied
7	Видимість файлу Windows → Linux	Windows VM / UbuntuClient	SMB/NFS	Файл видно на Linux	Успішно
8	Видимість файлу Linux → Windows	UbuntuClient / Windows VM	NFS/SMB	Файл видно у Windows	Успішно
9	Перевірка одночасного доступу	Windows VM + UbuntuClient	SMB + NFS	Зміни видно з обох сторін	Успішно
10	Перевірка SMB-блокувань	FileServer	SMB	Відображається lease/lock	Успішно

За результатами функціонального тестування встановлено, що файловий сервер змішаного доступу працює коректно. Windows-клієнт успішно отримує доступ до спільного ресурсу через SMB, Linux-клієнт успішно монтує той самий каталог через NFS, файли, створені одним клієнтом, відображаються на іншому, а

права користувачів ivan і petro застосовуються відповідно до розробленої моделі доступу. Це підтверджує працездатність реалізованої системи на функціональному рівні.

3.4 Навантажувальне тестування продуктивності (методика, інструменти, швидкість читання/запису під навантаженням)

Після завершення функціонального тестування було виконано навантажувальне тестування продуктивності файлового сервера змішаного доступу. Метою цього етапу є оцінювання швидкості передавання даних у лабораторній мережі, а також практичної швидкості читання і запису файлів через протоколи NFS та SMB. Таке тестування дозволяє визначити, наскільки ефективно працює створена система в умовах передавання великих файлів між клієнтами та сервером.

Навантажувальне тестування виконувалося у три основні етапи. На першому етапі було перевірено пропускну здатність мережевого каналу між UbuntuClient і FileServer за допомогою утиліти iperf3. На другому етапі було перевірено швидкість запису та читання через NFS за допомогою утиліти dd. На третьому етапі було перевірено швидкість запису та читання через SMB з боку Windows-клієнта за допомогою robocopy.

Першим було виконано тестування пропускну здатності мережі. Для цього на FileServer було запущено iperf3 у режимі сервера командою iperf3 -s. На UbuntuClient виконувалося підключення до сервера за IP-адресою 192.168.56.10. Тест тривав 30 секунд, що дозволило отримати більш стабільний результат, ніж короткий одноразовий запуск.

Команда для прямого тесту мережі мала вигляд:

iperf3 -c 192.168.56.10 -t 30

У результаті було отримано середню швидкість передавання даних приблизно 2.30 Gbit/s у напрямку UbuntuClient → FileServer. Цей результат показує, що віртуальна Host-only мережа забезпечує достатню пропускну здатність для подальшого тестування файлових протоколів.

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		76

```

Processing triggers for libc-bin (2.43-2ubuntu2)...
administrator@administrator-VirtualBox:~$ iperf3 -c 192.168.56.10 -t 30
Connecting to host 192.168.56.10, port 5201
[ 5] local 192.168.56.20 port 50226 connected to 192.168.56.10 port 5201
[ ID] Interval           Transfer     Bitrate      Retr  Cwnd
[ 5]  0.00-1.00   sec    248 MBytes  2.08 Gbits/sec  492   230 KBytes
[ 5]  1.00-2.00   sec    265 MBytes  2.22 Gbits/sec  417   287 KBytes
[ 5]  2.00-3.00   sec    289 MBytes  2.42 Gbits/sec  415   311 KBytes
[ 5]  3.00-4.00   sec    264 MBytes  2.21 Gbits/sec  507   230 KBytes
[ 5]  4.00-5.00   sec    265 MBytes  2.23 Gbits/sec  440   205 KBytes
[ 5]  5.00-6.00   sec    260 MBytes  2.18 Gbits/sec  454   240 KBytes
[ 5]  6.00-7.00   sec    270 MBytes  2.26 Gbits/sec  398   222 KBytes
[ 5]  7.00-8.00   sec    283 MBytes  2.38 Gbits/sec  597   288 KBytes
[ 5]  8.00-9.00   sec    317 MBytes  2.66 Gbits/sec  672   238 KBytes
[ 5]  9.00-10.00  sec    320 MBytes  2.68 Gbits/sec  534   185 KBytes
[ 5] 10.00-11.00  sec    286 MBytes  2.40 Gbits/sec  538   191 KBytes
[ 5] 11.00-12.00  sec    304 MBytes  2.55 Gbits/sec  506   277 KBytes
[ 5] 12.00-13.00  sec    243 MBytes  2.04 Gbits/sec  337   270 KBytes
[ 5] 13.00-14.00  sec    261 MBytes  2.19 Gbits/sec  414   216 KBytes
[ 5] 14.00-15.00  sec    272 MBytes  2.28 Gbits/sec  501   187 KBytes
[ 5] 15.00-16.00  sec    281 MBytes  2.35 Gbits/sec  391   304 KBytes
[ 5] 16.00-17.00  sec    240 MBytes  2.02 Gbits/sec  435   189 KBytes
[ 5] 17.00-18.00  sec    239 MBytes  2.00 Gbits/sec  429   188 KBytes
[ 5] 18.00-19.00  sec    297 MBytes  2.49 Gbits/sec  477   189 KBytes
[ 5] 19.00-20.00  sec    298 MBytes  2.50 Gbits/sec  654   229 KBytes
[ 5] 20.00-21.00  sec    268 MBytes  2.25 Gbits/sec  410   245 KBytes
[ 5] 21.00-22.00  sec    308 MBytes  2.58 Gbits/sec  601   269 KBytes
[ 5] 22.00-23.00  sec    290 MBytes  2.43 Gbits/sec  562   320 KBytes
[ 5] 23.00-24.00  sec    377 MBytes  3.17 Gbits/sec  577   287 KBytes
[ 5] 24.00-25.00  sec    179 MBytes  1.50 Gbits/sec  180   222 KBytes
[ 5] 25.00-26.00  sec    224 MBytes  1.88 Gbits/sec  375   194 KBytes
[ 5] 26.00-27.00  sec    222 MBytes  1.86 Gbits/sec  360   252 KBytes
[ 5] 27.00-28.00  sec    279 MBytes  2.34 Gbits/sec  436   236 KBytes
[ 5] 28.00-29.00  sec    257 MBytes  2.16 Gbits/sec  442   253 KBytes
[ 5] 29.00-30.00  sec    325 MBytes  2.72 Gbits/sec  697   197 KBytes
-----
[ ID] Interval           Transfer     Bitrate      Retr
[ 5]  0.00-30.00  sec   8.04 GBytes  2.30 Gbits/sec  14248
[ 5]  0.00-30.00  sec   8.04 GBytes  2.30 Gbits/sec
sender
receiver

```

Рисунок 3.24 — Результат тестування мережі iperf3 у напрямку UbuntuClient → FileServer

Для перевірки зворотного напрямку було використано параметр -R. У цьому режимі сервер передає дані клієнту, а клієнт приймає їх. Це дозволяє оцінити пропускну здатність у напрямку FileServer → UbuntuClient.

Команда для зворотного тесту мала вигляд:

iperf3 -c 192.168.56.10 -t 30 -R

У результаті було отримано середню швидкість приблизно 2.37 Gbit/s у напрямку FileServer → UbuntuClient. Отримані значення прямого і зворотного тестів є близькими, що свідчить про стабільну роботу мережевого каналу між клієнтом і сервером.

```

[ ID] Interval          Transfer    Bitrate      Retr
[ 5]  0.00-30.01   sec  8.29 GBytes  2.37 Gbits/sec  8333
[ 5]  0.00-30.00   sec  8.29 GBytes  2.37 Gbits/sec
iperf Done.
administrator@administrator-VirtualBox:~$

```

Рисунок 3.25 — Результат тестування мережі iperf3 у напрямку FileServer → UbuntuClient

Наступним етапом стало тестування продуктивності NFS. Оскільки UbuntuClient монтує каталог FileServer як /mnt/mixed, тест запису виконувався шляхом створення файлу безпосередньо в цьому каталозі. Для цього використовувалася команда dd, яка дозволяє створювати файл заданого розміру і вимірювати швидкість запису.

Команда для тестування запису через NFS мала вигляд:

```
sudo -u ivan dd if=/dev/zero of=/mnt/mixed/dd_nfs_test.dat bs=1M count=64 conv=fdatasync status=progress
```

У цьому тесті було створено файл розміром 64 MiB. Параметр bs=1M задає розмір блока 1 MiB, count=64 визначає кількість блоків, а conv=fdatasync забезпечує скидання даних на диск перед завершенням команди. Це дозволяє отримати більш реалістичний результат запису, ніж звичайне кешоване копіювання.

За результатами тестування швидкість запису через NFS становила приблизно 186 MB/s. Такий результат є достатнім для лабораторного стенду та підтверджує, що NFS-доступ працює не лише функціонально, а й забезпечує прийнятну швидкість передавання даних.

```

administrator@administrator-VirtualBox:~$ sudo -u ivan dd if=/dev/zero of=/mnt/mixed
/dd_nfs_test.dat bs=1M count=64 conv=fdatasync status=progress
64+0 records in
64+0 records out
67108864 bytes (67 MB, 64 MiB) copied, 0.360659 s, 186 MB/s
administrator@administrator-VirtualBox:~$

```

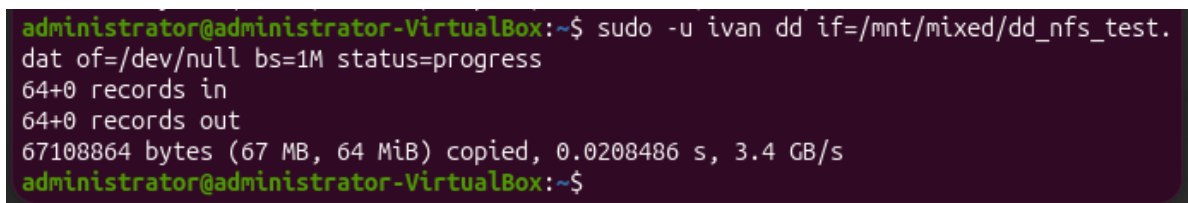
Рисунок 3.26 — Результат тестування швидкості запису через NFS за допомогою dd

Для перевірки читання через NFS було виконано зчитування створеного файлу в /dev/null. Такий підхід дозволяє виміряти швидкість читання без запису результату на локальний диск клієнта.

Команда для тестування читання через NFS мала вигляд:

sudo -u ivan dd if=/mnt/mixed/dd_nfs_test.dat of=/dev/null bs=1M status=progress

У результаті було отримано швидкість читання приблизно 3.4 GB/s. Таке значення є значно вищим за швидкість запису та пропускну здатність мережі, тому його необхідно інтерпретувати з урахуванням кешування операційної системи. Оскільки файл зчитувався невдовзі після створення, частина даних могла перебувати в кеші оперативної пам'яті, що суттєво підвищило показник читання.



```
administrator@administrator-VirtualBox:~$ sudo -u ivan dd if=/mnt/mixed/dd_nfs_test.  
dat of=/dev/null bs=1M status=progress  
64+0 records in  
64+0 records out  
67108864 bytes (67 MB, 64 MiB) copied, 0.0208486 s, 3.4 GB/s  
administrator@administrator-VirtualBox:~$
```

Рисунок 3.27 — Результат тестування швидкості читання через NFS за допомогою dd

Третім етапом стало тестування продуктивності SMB з боку Windows-клієнта. Для цього використовувалася утиліта robocopy, яка є стандартним засобом Windows для копіювання файлів і дозволяє отримати швидкість передавання даних. Спочатку на локальному диску Windows було створено тестовий файл розміром 256 MiB за допомогою fsutil.

Команда створення тестового файлу:

fsutil file createnew C:\smb_test_256mb.bin 268435456

Після цього файл було скопійовано з Windows VM на SMB-ресурс, підключений як диск Z:. Для копіювання використовувалася команда robocopy з параметром /J, який вмикає режим некешованого введення-виведення для великих файлів.

Команда тестування SMB-запису:

robocopy C:\Users|Maksim\Desktop Z:\ smb_test_256mb.bin /J /NP

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		79

У результаті було отримано швидкість запису приблизно 142860806 Bytes/sec, що відповідає приблизно 143 MB/s. Це значення показує фактичну швидкість передавання файлу з Windows-клієнта на файловий сервер через SMB.

```
C:\Windows\system32>fsutil file createnew C:\smb_test_256mb.bin 268435456
File C:\smb_test_256mb.bin is created

C:\Windows\system32>robocopy C:\Users\Maksim\Desktop Z:\ smb_test_256mb.bin /J /NP

-----
ROBOCOPY      ::      Robust File Copy for Windows
-----

Started : 20 червня 2026 р. 20:46:17
Source  : C:\Users\Maksim\Desktop\
Dest    : Z:\

Files : smb_test_256mb.bin

Options : /DCOPY:DA /COPY:DAT /J /NP /R:1000000 /W:30

-----

New File      1      C:\Users\Maksim\Desktop\
                256.0 m      smb_test_256mb.bin

-----

      Total    Copied    Skipped    Mismatch    FAILED    Extras
Dirs  :        1         0         1         0         0         0
Files :        1         1         0         0         0         0
Bytes :   256.00 m   256.00 m         0         0         0         0
Times :    0:00:01    0:00:01                0:00:00    0:00:00

Speed :           142860806 Bytes/sec.
Speed :           8174.560 MegaBytes/min.
Ended : 20 червня 2026 р. 20:46:19

C:\Windows\system32>
```

Рисунок 3.28 — Результат тестування швидкості запису через SMB за допомогою robocopy

Для перевірки SMB-читання файл було скопійовано у зворотному напрямку — з мережевого диска Z: на локальний диск Windows у каталог C:\SMBDownload. Це дозволило оцінити швидкість передавання даних від FileServer до Windows VM через SMB.

Команда тестування SMB-читання:

robocopy Z:\ C:\SMBDownload smb_test_256mb.bin /J /NP

У результаті було отримано швидкість приблизно 108678322 Bytes/sec, що відповідає приблизно 109 MB/s. Цей результат є нижчим за швидкість SMB-запису,

але залишається прийнятним для лабораторного середовища та підтверджує працездатність SMB-доступу під час передавання великих файлів.

```
C:\Windows\system32>mkdir C:\SMBDownload

C:\Windows\system32>robocopy Z:\ C:\SMBDownload smb_test_256mb.bin /J /NP

-----
ROBOCOPY      ::      Robust File Copy for Windows
-----

Started : 20 червня 2026 р. 20:47:27
Source  : Z:\
Dest    : C:\SMBDownload\

Files   : smb_test_256mb.bin

Options : /DCOPY:DA /COPY:DAT /J /NP /R:1000000 /W:30

-----

New File      1      Z:\
              256.0 m      smb_test_256mb.bin

-----

Total      Copied      Skipped      Mismatch      FAILED      Extras
 Dirs  :      1          0          1          0          0          0
Files  :      1          1          0          0          0          0
Bytes  : 256.00 m 256.00 m          0          0          0          0
Times  : 0:00:02 0:00:02          0          0          0          0

Speed :      108678322 Bytes/sec.
Speed :      6218.623 MegaBytes/min.
Ended : 20 червня 2026 р. 20:47:30

C:\Windows\system32>
```

Рисунок 3.29 — Результат тестування швидкості читання через SMB за допомогою robocopy

Результати навантажувального тестування наведено в таблиці 3.2.

Таблиця 3.2

Результати навантажувального тестування продуктивності

№	Вид тесту	Інструмент	Напрямок передавання	Результат
1	Пропускна здатність мережі	iperf3	UbuntuClient → FileServer	2.30 Gbit/s
2	Пропускна здатність мережі	iperf3 -R	FileServer → UbuntuClient	2.37 Gbit/s
3	NFS запис	dd	UbuntuClient → FileServer	186 MB/s
4	NFS читання	dd	FileServer → UbuntuClient	3.4 GB/s
5	SMB запис	robocopy	Windows VM → FileServer	143 MB/s
6	SMB читання	robocopy	FileServer → Windows VM	109 MB/s

Аналіз отриманих результатів показує, що мережева пропускна здатність у лабораторному середовищі є достатньою для тестування файлових протоколів. Значення `iperf3` у прямому та зворотному напрямках перевищують 2 Gbit/s, що свідчить про стабільну роботу Host-only мережі між віртуальними машинами.

Результати NFS-запису показали швидкість 186 MB/s, що є добрим показником для лабораторного стенду. Значення NFS-читання 3.4 GB/s слід розглядати з урахуванням кешування операційної системи, оскільки файл зчитувався після попереднього запису та міг перебувати в оперативній пам'яті. Тому цей показник не можна прямо порівнювати з реальною дисковою або мережевою швидкістю без додаткового очищення кешу.

Результати SMB-тестування показали швидкість запису приблизно 143 MB/s і швидкість читання приблизно 109 MB/s. Ці значення нижчі за пропускну здатність, отриману через `iperf3`, що є очікуваним, оскільки файлові протоколи мають додаткові витрати на автентифікацію, обробку файлових операцій, роботу з правами доступу та взаємодію з файловою системою.

Таким чином, навантажувальне тестування підтвердило, що реалізований файловий сервер змішаного доступу забезпечує стабільну передачу даних через SMB і NFS. Отримані результати є достатніми для лабораторного стенду та демонструють працездатність системи не лише на рівні підключення, а й під час передавання файлів значного розміру.

3.5 Тестування безпеки та відмовостійкості

Після функціонального та навантажувального тестування було виконано перевірку базових механізмів безпеки й відмовостійкості файлового сервера змішаного доступу. Метою цього етапу є підтвердження того, що сервер відкриває лише необхідні мережеві служби, фіксує невдалі спроби автентифікації, коректно обмежує права користувачів, не надає надмірних привілеїв NFS-клієнтам і відновлює роботу після перезавантаження.

Оскільки система працює в ізольованій Host-only мережі 192.168.56.0/24, ризик зовнішнього доступу до стенду є обмеженим. Проте навіть у лабораторному

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		82

середовищі важливо перевірити, які служби доступні на сервері, чи відповідають вони призначенню системи та чи не відкриті зайві порти. Для цього було виконано сканування FileServer за допомогою утиліти nmap з боку UbuntuClient.

Команда сканування мала вигляд:

nmap -sV 192.168.56.10

У результаті сканування було виявлено відкриті порти, що відповідають службам, необхідним для роботи файлового сервера: SSH для віддаленого адміністрування, rpcbind і NFS для доступу Linux-клієнта, а також Samba/SMB для доступу Windows-клієнта. Інші перевірені TCP-порти перебували у закритому стані. Це підтверджує, що на сервері доступні лише служби, необхідні для виконання поставлених завдань.

```

administrator@administrator-VirtualBox: ~
administrator@administrator-VirtualBox:~$ nmap -sV 192.168.56.10
Starting Nmap 7.98 ( https://nmap.org ) at 2026-06-21 15:25 +0300
Nmap scan report for 192.168.56.10
Host is up (0.0099s latency).
Not shown: 995 closed tcp ports (conn-refused)
PORT      STATE SERVICE        VERSION
22/tcp    open  ssh            OpenSSH 10.2p1 Ubuntu 2ubuntu3.2 (Ubuntu Linux; protocol 2.0)
111/tcp   open  rpcbind        2-4 (RPC #100000)
139/tcp   open  netbios-ssn    Samba smbd 4
445/tcp   open  netbios-ssn    Samba smbd 4
2049/tcp  open  nfs            3-4 (RPC #100003)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org
/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.86 seconds
administrator@administrator-VirtualBox:~$
  
```

Рисунок 3.30 — Результат сканування відкритих портів FileServer за допомогою nmap

Відкритий порт 22/tcp відповідає службі SSH, яка використовується для віддаленого адміністрування FileServer. Порти 139/tcp і 445/tcp відповідають службам Samba/SMB, які забезпечують доступ Windows-клієнтів до спільного ресурсу mixed. Порт 2049/tcp використовується NFS-сервером, а порт 111/tcp

відповідає rpcbind, який потрібний для роботи NFS-служб. Такий набір відкритих портів відповідає функціональному призначенню розгорнутого файлового сервера.

Наступним етапом було перевірено журналювання невдалих спроб автентифікації через SSH. Для цього з клієнтської машини було виконано спробу підключення до FileServer під неіснуючим користувачем fakeuser. Після введення неправильного пароля на FileServer було переглянуто системний журнал служби SSH за допомогою journalctl.

Команда перегляду журналу:

sudo journalctl -u ssh --since "10 minutes ago" --no-pager

У журналі було зафіксовано записи типу Invalid user fakeuser та Failed password for invalid user fakeuser. Це підтверджує, що сервер реєструє невдалі спроби входу та дозволяє адміністратору виявляти потенційні спроби несанкціонованого доступу.

```
administrator@fileserver:~$ sudo journalctl -u ssh --since "10 minutes ago" --no-pager
чер 20 18:27:36 fileserver sshd-session[2378]: Connection closed by 192.168.56.20 port 42920
чер 20 18:29:40 fileserver sshd-session[2382]: Invalid user fakeuser from 192.168.56.30 port 51510
чер 20 18:29:51 fileserver sshd-session[2382]: pam_unix(sshd:auth): check pass; user unknown
чер 20 18:29:51 fileserver sshd-session[2382]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.56.30
чер 20 18:29:53 fileserver sshd-session[2382]: Failed password for invalid user fakeuser from 192.168.56.30 port 51510 ssh2
чер 20 18:30:02 fileserver sshd-session[2382]: pam_unix(sshd:auth): check pass; user unknown
чер 20 18:30:04 fileserver sshd-session[2382]: Failed password for invalid user fakeuser from 192.168.56.30 port 51510 ssh2
чер 20 18:30:11 fileserver sshd-session[2382]: Connection reset by invalid user fakeuser 192.168.56.30 port 51510 [preauth]
чер 20 18:30:11 fileserver sshd-session[2382]: PAM 1 more authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.56.30
administrator@fileserver:~$
```

Рисунок 3.31 — Журналювання невдалої SSH-спроби входу користувача fakeuser

Журналювання подій автентифікації є важливим елементом безпеки, оскільки дозволяє аналізувати спроби підбору паролів, виявляти підозрілу активність і контролювати доступ до сервера. У реальному середовищі такі журнали можуть додатково передаватися до централізованої системи моніторингу або аналізуватися засобами виявлення вторгнень.

Окремо було перевірено обмеження прав користувача petro. Відповідно до розробленої моделі доступу, користувач petro має лише права читання до спільного каталогу /srv/storage/mixed і не повинен мати можливості створювати або

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				84
Змн.	Арк.	№ докум.	Підпис	Дата		

змінювати файли. Для перевірки цього обмеження на UbuntuClient було виконано команду створення файлу від імені petro в каталозі /mnt/mixed.

Команда перевірки:

sudo -u petro touch /mnt/mixed/petro_security_test.txt

Очікуваним результатом є повідомлення Permission denied. Отримана відмова підтверджує, що користувач petro не має прав запису через NFS і не може створювати файли у спільному каталозі.

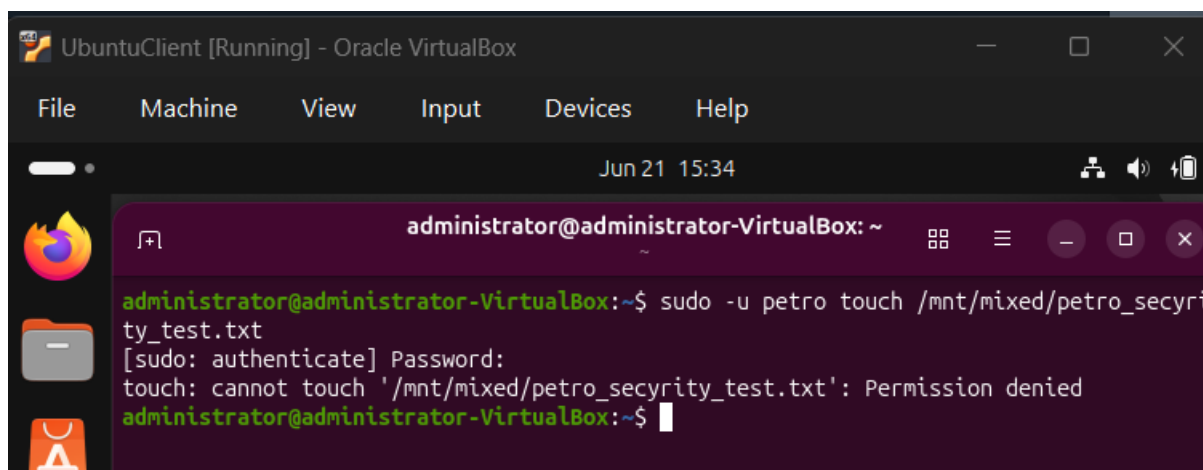


Рисунок 3.32 — Відмова у створенні файлу користувачем petro через NFS

Такий результат є важливим, оскільки підтверджує коректність налаштування POSIX ACL. Обмеження застосовується на рівні серверної файлової системи, тому воно діє незалежно від того, через який протокол користувач звертається до файлів. Це означає, що права доступу не дублюються окремо для кожного протоколу, а централізовано контролюються Linux-сервером.

Додатково було перевірено роботу механізму root_squash для NFS. Цей механізм призначений для того, щоб користувач root на клієнтській машині не отримував автоматично root-права на серверному файловому ресурсі. Без такого обмеження адміністратор клієнтської машини міг би отримати надмірні привілеї над файлами на сервері, що створювало б ризик порушення безпеки.

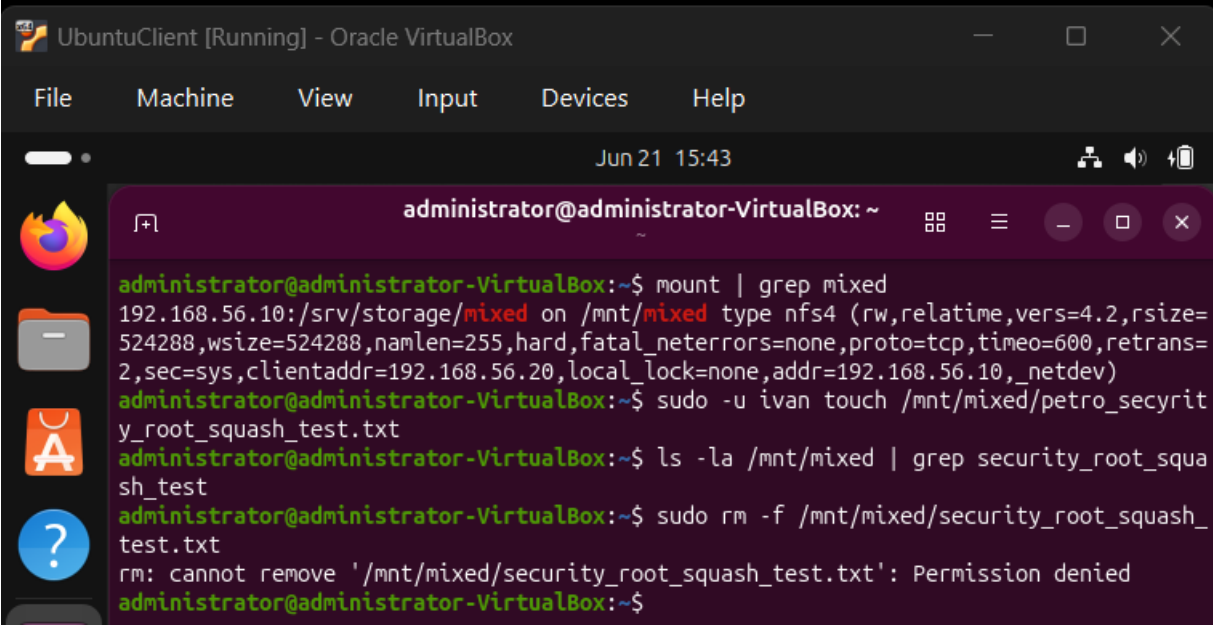
У межах тестування було створено файл від імені користувача ivan, після чого виконано спробу видалення або зміни цього файлу через sudo з боку UbuntuClient. Через дію root_squash така операція не повинна надавати клієнтському root повні права на серверний ресурс. У разі відсутності відповідних прав операція завершується повідомленням Permission denied.

Команди перевірки:

```
sudo -u ivan touch /mnt/mixed/security_root_squash_test.txt
```

```
sudo rm -f /mnt/mixed/security_root_squash_test.txt
```

Очікуваний результат для другої команди — відмова в доступі або неможливість виконання операції без відповідних серверних прав. Це підтверджує, що root на NFS-клієнті не прирівнюється до root на FileServer.



```
administrator@administrator-VirtualBox: ~
administrator@administrator-VirtualBox:~$ mount | grep mixed
192.168.56.10:/srv/storage/mixed on /mnt/mixed type nfs4 (rw,relatime,vers=4.2,rsize=
524288,wsz=524288,namlen=255,hard,fatal_nererrors=none,proto=tcp,timeo=600,retrans=
2,sec=sys,clientaddr=192.168.56.20,local_lock=none,addr=192.168.56.10,_netdev)
administrator@administrator-VirtualBox:~$ sudo -u ivan touch /mnt/mixed/petro_secryit
y_root_squash_test.txt
administrator@administrator-VirtualBox:~$ ls -la /mnt/mixed | grep security_root_squa
sh_test
administrator@administrator-VirtualBox:~$ sudo rm -f /mnt/mixed/security_root_squash
test.txt
rm: cannot remove '/mnt/mixed/security_root_squash_test.txt': Permission denied
administrator@administrator-VirtualBox:~$
```

Рисунок 3.33 — Перевірка роботи механізму root_squash для NFS

Наступним етапом було перевірено відмовостійкість системи на базовому рівні. У межах лабораторного стенду під відмовостійкістю розглядається здатність файлового сервера та клієнтів відновлювати роботу після перезавантаження. Для цього було перезавантажено FileServer, UbuntuClient і Windows VM, після чого повторно перевірено IP-адресацію, стан служб і доступність спільного каталогу.

На FileServer після перезавантаження було перевірено стан служб SSH, Samba та NFS. Для цього використовувалися команди:

```
systemctl status ssh --no-pager
```

```
systemctl status smbd --no-pager
```

```
systemctl status nfs-server --no-pager
```

Очікуваним результатом є активний стан служб. Служба ssh повинна бути active (running), служба smbd — active (running), а служба nfs-server може відображатися як active (exited), що є нормальним для NFS-сервера в Linux. Усі ці служби повинні бути enabled, тобто запускатися автоматично після перезавантаження системи.

```

FileServ [Running] - Oracle VirtualBox
File Machine View Input Devices Help

administrator@fileserv:~$ systemctl status ssh --no-pager
• ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enabled)
  Active: active (running) since Sun 2026-06-21 08:56:54 UTC; 3h 51min ago
  Invocation: b9cb235007124eb4b86d61566a37d657
  TriggeredBy: • ssh.socket
  Docs: man:sshd(8)
        man:sshd_config(5)
  Process: 1329 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
  Main PID: 1368 (sshd)
  Tasks: 1 (limit: 1713)
  Memory: 2.5M (peak: 4.2M)
  CPU: 126ms
  CGroup: /system.slice/ssh.service
          └─1368 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

чеп 21 08:56:54 fileserv systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
чеп 21 08:56:54 fileserv sshd[1368]: Server listening on 0.0.0.0 port 22.
чеп 21 08:56:54 fileserv systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
чеп 21 08:56:54 fileserv sshd[1368]: Server listening on :: port 22.
чеп 21 12:25:09 fileserv sshd-session[2165]: Connection closed by 192.168.56.20 port 53326
administrator@fileserv:~$ _

```

Рисунок 3.34 — Стан служб SSH після перезавантаження FileServer

```

FileServ [Running] - Oracle VirtualBox
File Machine View Input Devices Help

administrator@fileserv:~$ systemctl status smb --no-pager
• smbd.service - Samba SMB Daemon
  Loaded: loaded (/usr/lib/systemd/system/smbd.service; enabled; preset: enabled)
  Active: active (running) since Sun 2026-06-21 08:56:55 UTC; 3h 51min ago
  Invocation: 2a928e54f13d4fddb1fca430fb112a2a
  Docs: man:smbd(8)
        man:samba(7)
        man:smb.conf(5)
  Process: 1501 ExecCondition=/usr/share/samba/is-configured smb (code=exited, status=0/SUCCESS)
  Process: 1515 ExecStartPre=/usr/share/samba/update-apparmor-samba-profile (code=exited, status=0/SUCCESS)
  Main PID: 1541 (smbd)
  Status: "smbd: ready to serve connections..."
  Tasks: 4 (limit: 1713)
  Memory: 20.9M (peak: 49M)
  CPU: 2.112s
  CGroup: /system.slice/smbd.service
          └─1541 /usr/sbin/smbd --foreground --no-process-group
            └─1553 "smbd: notifyd"
              └─1554 "smbd: cleanupd"
                └─1749 "smbd: client [192.168.56.1]"

чеп 21 08:56:55 fileserv systemd[1]: Starting smbd.service - Samba SMB Daemon...
чеп 21 08:56:55 fileserv systemd[1]: Started smbd.service - Samba SMB Daemon.
administrator@fileserv:~$

```

Рисунок 3.35 — Стан служб Samba після перезавантаження FileServer

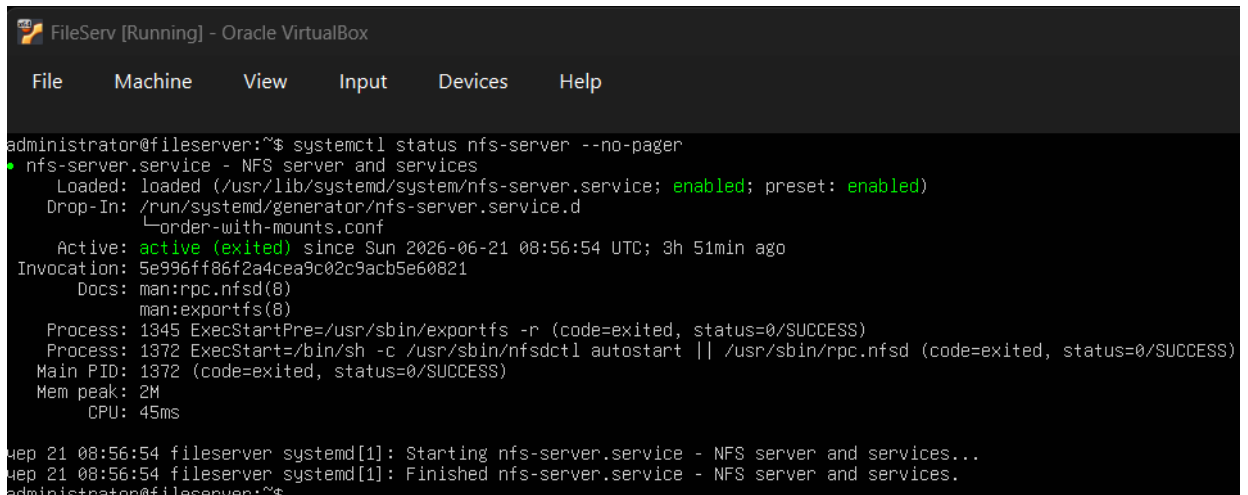


Рисунок 3.36 — Стан служб NFS після перезавантаження FileServer

На UbuntuClient після перезавантаження було перевірено автоматичне монтування NFS-ресурсу. Для цього використовувалася команда:

mount | grep mixed

Очікуваним результатом є наявність запису про монтування ресурсу 192.168.56.10:/srv/storage/mixed у каталог /mnt/mixed. Це підтверджує, що запис у /etc/fstab працює коректно, а NFS-ресурс підключається після запуску клієнтської системи.

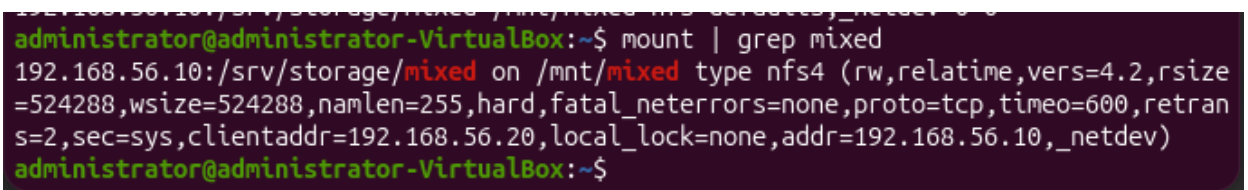


Рисунок 3.37 — Перевірка автоматичного NFS-монтування після перезавантаження UbuntuClient

Також було перевірено доступ Windows VM до SMB-ресурсу після перезавантаження. Windows-клієнт успішно підключився до ресурсу \\192.168.56.10\\mixed, що підтвердило працездатність Samba після повторного запуску сервера. Таким чином, система зберігає налаштування та відновлює роботу після перезавантаження.

Результати тестування безпеки та відмовостійкості узагальнено в таблиці 3.3.

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				88
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 3.3

Результати тестування безпеки та відмовостійкості

№	Перевірка	Інструмент / команда	Очікуваний результат	Фактичний результат
1	Сканування відкритих портів	nmap -sV 192.168.56.10	Відкриті лише потрібні служби	Успішно
2	Перевірка SSH-журналу	journalctl -u ssh	Фіксація невдалої спроби входу	Успішно
3	Обмеження прав petro через NFS	sudo -u petro touch	Permission denied	Успішно
4	Обмеження прав petro через SMB	створення файлу у Windows	Access is denied	Успішно
5	Перевірка root_squash	sudo-операція на NFS-ресурсі	Немає надмірних root-прав	Успішно
6	Автозапуск SSH	systemctl status ssh	Служба активна	Успішно
7	Автозапуск Samba	systemctl status smbd	Служба активна	Успішно
8	Автозапуск NFS	systemctl status nfs-server	Служба активна	Успішно
9	Автомонтування NFS	mount grep mixed	Ресурс змонтовано	Успішно
10	Доступ SMB після reboot	\\192.168.56.10\\mixed	Ресурс відкривається	Успішно

Проведене тестування показало, що файловий сервер змішаного доступу має базовий рівень захищеності, достатній для лабораторного стенду. На сервері відкриті лише служби, необхідні для його роботи. Невдалі SSH-спроби входу фіксуються в системному журналі. Користувач із правами лише читання не може створювати файли у спільному каталозі. Механізм root_squash обмежує надмірні привілеї root-користувача NFS-клієнта. Після перезавантаження серверні служби та NFS-монтування відновлюють роботу.

Таким чином, результати тестування безпеки та відмовостійкості підтверджують, що реалізована система не лише забезпечує змішаний доступ SMB/NFS, а й має базові механізми контролю доступу, журналювання та відновлення працездатності після перезапуску.

Висновки до розділу 3

У третьому розділі було виконано практичну розробку та тестування файлового сервера змішаного доступу SMB/NFS на базі Linux. Практична реалізація здійснювалася у віртуалізованому лабораторному середовищі, яке складалося з трьох основних вузлів: FileServer на базі Ubuntu Server, UbuntuClient на базі Ubuntu Desktop та Windows VM на базі Windows. Для взаємодії між вузлами було використано ізольовану Host-only мережу 192.168.56.0/24.

На першому етапі було розгорнуто серверну операційну систему Ubuntu Server, налаштовано мережеву адресу FileServer 192.168.56.10, підготовлено дискову структуру та створено спільний каталог /srv/storage/mixed. Для організації файлового простору було створено підкаталоги archive, projects і public, які використовувалися для демонстрації структури спільного ресурсу та перевірки доступу з клієнтських систем.

Було створено локальних користувачів і групи для реалізації моделі розмежування доступу. Користувач ivan отримав права читання та запису, користувач retro — лише права читання. Для групового керування доступом використовувалися групи fs_rw, fs_ro, fs_admins та fs_data. За допомогою POSIX ACL було налаштовано права доступу до каталогу /srv/storage/mixed, а також default ACL для успадкування прав новими файлами та каталогами.

Для забезпечення доступу Windows-клієнта було встановлено та налаштовано службу Samba. У конфігураційному файлі /etc/samba/smb.conf створено SMB-ресурс mixed, який вказує на каталог /srv/storage/mixed. Після налаштування було перевірено стан служби smbd, яка перебувала в активному стані та була готова приймати підключення від Windows-клієнтів. Windows VM успішно отримала доступ до спільного ресурсу за шляхом \\192.168.56.10\\mixed.

Для забезпечення доступу Linux-клієнта було налаштовано NFS-сервер. У файлі /etc/exports каталог /srv/storage/mixed було експортовано для мережі 192.168.56.0/24. На UbuntuClient ресурс було змонтовано в каталог /mnt/mixed, а також налаштовано автоматичне монтування через /etc/fstab. Перевірка командою

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		90

mount | grep mixed підтвердила, що NFS-ресурс успішно підключений до клієнтської системи.

У межах функціонального тестування було перевірено доступ до спільного каталогу з Windows- та Linux-клієнтів. Файли, створені через SMB у Windows VM, відображалися на UbuntuClient через NFS. У зворотному напрямку файли, створені через NFS на UbuntuClient, були видимі у Windows через SMB. Це підтвердило, що обидва протоколи працюють з одним і тим самим серверним каталогом /srv/storage/mixed.

Було перевірено права доступу користувачів. Користувач ivan успішно створював файли у спільному каталозі, що підтвердило наявність прав читання та запису. Користувач retro під час спроби створення файлу отримував повідомлення Permission denied у Linux або Access is denied у Windows, що підтвердило правильність налаштування обмежень доступу. Також було перевірено сценарій одночасного змішаного доступу до файлу concurrent_test.txt, у межах якого зміни, внесені через один протокол, ставали видимими через інший.

Під час перевірки механізмів блокування було використано команду smbstatus --locks. Вона показала активні SMB-блокування або lease для відкритого файлу, що підтвердило можливість контролю стану відкритих файлів з боку Samba. Це є важливим для аналізу поведінки файлового сервера під час роботи Windows-клієнтів зі спільними файлами.

Навантажувальне тестування показало, що пропускна здатність мережі між UbuntuClient і FileServer становить приблизно 2.30 Gbit/s у напрямку UbuntuClient → FileServer та 2.37 Gbit/s у зворотному напрямку. Тестування NFS за допомогою dd показало швидкість запису приблизно 186 MB/s і швидкість читання приблизно 3.4 GB/s. Високе значення читання NFS пояснюється впливом кешування операційної системи. Тестування SMB за допомогою robocopy показало швидкість запису приблизно 143 MB/s і швидкість читання приблизно 109 MB/s.

У межах тестування безпеки було виконано сканування FileServer за допомогою nmap. Результати показали, що відкритими є лише порти служб, необхідних для роботи системи: SSH, rpcbind, Samba/SMB та NFS. Також було перевірено

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		91

журналювання невдалих SSH-спроб входу. Під час спроби підключення під неіснуючим користувачем fakeuser у журналі ssh.service було зафіксовано події Invalid user та Failed password, що підтвердило роботу механізму журналювання автентифікації.

Додатково було перевірено роботу механізму root_squash для NFS. Під час спроби виконання операцій із файлами через sudo на NFS-клієнті було підтверджено, що root на клієнтській машині не отримує автоматично root-права на серверному файловому ресурсі. Це є важливим елементом безпеки NFS, оскільки він обмежує надмірні привілеї клієнтських систем.

Перевірка відмовостійкості показала, що після перезавантаження FileServer служби SSH, Samba та NFS відновлюють роботу. Також було перевірено доступність спільного ресурсу після перезапуску клієнтських систем. UbuntuClient повторно підключав NFS-ресурс, а Windows VM мала можливість знову отримати доступ до SMB-ресурсу. Це підтвердило збереження налаштувань і працездатність системи після перезавантаження.

Таким чином, у третьому розділі було повністю реалізовано та протестовано файловий сервер змішаного доступу SMB/NFS на базі Linux. Проведені тести підтвердили працездатність Samba і NFS, коректність моделі прав доступу, можливість змішаного доступу з Windows- та Linux-клієнтів, прийнятну продуктивність передавання даних, базовий рівень безпеки та здатність системи відновлювати роботу після перезавантаження.

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		92

ВИСНОВКИ

У ході виконання кваліфікаційної роботи на тему «Проект файлового сервера змішаного доступу SMB/NFS на базі Linux» було спроектовано, розгорнуто та протестовано файловий сервер, який забезпечує спільний доступ до одного серверного каталогу для клієнтів із різними операційними системами. Практична реалізація виконувалася у віртуалізованому лабораторному середовищі, що дозволило змодельовати гетерогенну мережу з Windows- та Linux-клієнтами.

У першому розділі було виконано аналіз предметної області та сформовано постановку задачі. Було розглянуто призначення файлових серверів у гетерогенних мережах, особливості протоколів SMB/CIFS і NFS, а також роль Samba як засобу надання SMB-доступу з Linux-сервера для Windows-клієнтів. Встановлено, що SMB/CIFS є зручним і типовим рішенням для Windows-середовища, тоді як NFS є природним механізмом підключення віддалених каталогів у Linux- та Unix-подібних системах.

Окрему увагу було приділено проблемам одночасного змішаного доступу до одного файлового ресурсу. Було визначено, що під час одночасної роботи SMB- та NFS-клієнтів із одним каталогом виникають питання узгодженості кешування, блокування файлів, видимості змін і контролю прав доступу. Також було проаналізовано відмінності між моделями прав доступу POSIX ACL та Windows ACL. Показано, що для файлового сервера на базі Linux центральною є модель Linux-прав і POSIX ACL, а Samba та NFS повинні працювати з нею узгоджено.

У межах аналізу аналогів було розглянуто TrueNAS, готові апаратні NAS-рішення та самостійну конфігурацію Linux-сервера з Samba і NFS. За результатами порівняння встановлено, що для кваліфікаційної роботи найбільш доцільним є ручне розгортання файлового сервера на базі Linux. Такий підхід забезпечує повний контроль над конфігурацією, дає можливість детально дослідити роботу SMB, NFS, POSIX ACL, користувачів, груп і мережевих служб, а також має високу навчальну цінність.

У другому розділі було виконано моделювання файлового сервера змішаного доступу. Як серверну операційну систему було обрано Ubuntu Server, оскільки вона

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		93

є стабільною, підтримує необхідні серверні служби, має зручну систему керування пакетами та добре підходить для розгортання Samba і NFS. Як файлову систему обрано ext4, оскільки вона є простою в адмініструванні, підтримує POSIX ACL і повністю відповідає потребам лабораторного стенду.

Було спроектовано архітектуру системи, що складається з трьох віртуальних машин: FileServer, UbuntuClient і Windows VM. FileServer виконує роль центрального файлового сервера з IP-адресою 192.168.56.10. UbuntuClient з IP-адресою 192.168.56.20 використовується для доступу до спільного каталогу через NFS. Windows VM з IP-адресою 192.168.56.30 використовується для перевірки доступу через SMB/CIFS. Взаємодія вузлів організована через ізольовану Host-only мережу 192.168.56.0/24.

Центральним спільним ресурсом системи визначено каталог /srv/storage/mixed. Саме цей каталог публікується через Samba для Windows-клієнта та експортується через NFS для Linux-клієнта. Така архітектура дозволяє організувати єдине файлове сховище, доступне різним операційним системам через відповідні для них протоколи.

Було розроблено модель прав доступу й автентифікації. У межах лабораторного стенду використано локальну модель користувачів Linux і Samba-облікових записів. Для перевірки прав було створено користувачів ivan і petro. Користувач ivan отримав права читання та запису, а користувач petro — лише права читання. Для групового керування доступом використано групи fs_rw, fs_ro, fs_admins та fs_data. Права до спільного каталогу було налаштовано за допомогою POSIX ACL і default ACL, що забезпечує коректне успадкування прав новими файлами та каталогами.

У третьому розділі було виконано практичну реалізацію системи. На FileServer було налаштовано мережеву адресу 192.168.56.10, створено структуру каталогів /srv/storage/mixed, встановлено необхідні пакети, налаштовано Samba, NFS, OpenSSH, користувачів, групи та права доступу. Для Windows-клієнта було створено SMB-ресурс mixed, доступний за шляхом \\192.168.56.10\\mixed. Для Linux-клієнта той самий каталог було змонтовано через NFS у точку /mnt/mixed.

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		94

Під час функціонального тестування було підтверджено доступність спільного ресурсу з обох клієнтських систем. Windows VM успішно відкривала SMB-ресурс, а UbuntuClient успішно монтував NFS-ресурс. Було перевірено, що файли, створені через Windows-клієнт, відображаються на Linux-клієнті, а файли, створені через Linux-клієнт, відображаються у Windows. Це підтвердило, що SMB і NFS працюють з одним і тим самим серверним каталогом, а не з різними копіями даних.

Перевірка прав доступу показала, що користувач ivan може створювати та змінювати файли у спільному каталозі, тоді як користувач petro не має права запису. Під час спроби створення файлу від імені petro система повертала повідомлення Permission denied у Linux або Access is denied у Windows. Це підтвердило правильність налаштування моделі прав доступу на основі груп і POSIX ACL.

Також було перевірено сценарій змішаного доступу та механізми блокування. Для аналізу відкритих SMB-файлів використовувалася команда smbstatus --locks, яка показала активні SMB-блокування або lease для відкритого файлу. Це підтвердило, що Samba відстежує стан відкритих файлів з боку Windows-клієнтів. Водночас було зазначено, що в реальних середовищах одночасне редагування одного файлу через SMB і NFS потребує обережності через відмінності механізмів кешування та блокування.

У межах навантажувального тестування було отримано практичні показники продуктивності системи. Пропускна здатність мережі між UbuntuClient і FileServer за результатами iperf3 становила приблизно 2.30 Gbit/s у напрямку UbuntuClient → FileServer і 2.37 Gbit/s у напрямку FileServer → UbuntuClient. Тестування NFS за допомогою dd показало швидкість запису приблизно 186 MB/s і швидкість читання приблизно 3.4 GB/s. При цьому високе значення читання NFS пояснюється впливом кешування операційної системи.

Тестування SMB за допомогою robocopy показало швидкість запису приблизно 143 MB/s і швидкість читання приблизно 109 MB/s. Отримані результати підтверджують, що реалізований файловий сервер забезпечує стабільну

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		95

передачу даних через обидва протоколи. Різниця між результатами `iperf3` і файловими тестами є очікуваною, оскільки SMB і NFS мають додаткові витрати на обробку файлових операцій, автентифікацію, перевірку прав доступу та взаємодію з файловою системою.

У межах тестування безпеки було виконано сканування FileServer за допомогою `ntmap`. Було встановлено, що відкритими є лише порти служб, необхідних для роботи системи: SSH, `rpcbind`, Samba/SMB та NFS. Також було перевірено журналювання невдалих SSH-спроб входу. Під час спроби підключення під неіснуючим користувачем `fakeuser` у журналі служби SSH було зафіксовано події `Invalid user` і `Failed password`. Це підтверджує, що сервер реєструє невдалі спроби автентифікації.

Додатково було перевірено роботу механізму `root_squash` у NFS, який не дозволяє користувачу `root` на клієнтській машині отримати повні `root`-права на серверному файловому ресурсі. Цей механізм є важливим елементом безпеки, оскільки обмежує надмірні привілеї клієнтських систем. Також було підтверджено, що після перезавантаження FileServer служби SSH, Samba та NFS відновлюють роботу, а UbuntuClient може повторно підключати NFS-ресурс.

Отже, поставлена мета кваліфікаційної роботи була досягнута. Було спроектовано та реалізовано файловий сервер змішаного доступу SMB/NFS на базі Linux, який забезпечує централізоване зберігання даних і доступ до спільного каталогу для Windows- та Linux-клієнтів. Реалізована система підтримує розмежування прав доступу, працює з єдиним спільним каталогом, забезпечує доступ через Samba і NFS, проходить функціональне, продуктивне та безпекове тестування.

Практична цінність роботи полягає в тому, що запропоноване рішення може бути використане як навчальний або лабораторний стенд для вивчення принципів роботи файлових серверів у гетерогенних мережах. Воно демонструє взаємодію SMB і NFS, налаштування Linux-прав і POSIX ACL, перевірку користувацьких ролей, тестування продуктивності та базові підходи до захисту файлового сервера.

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		96

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. The Official Samba Documentation. Samba Team : офіційна документація. URL: <https://www.samba.org/samba/docs/>.
2. Linux NFS-HOWTO. The Linux Documentation Project. URL: <https://nfs.sourceforge.net/nfs-howto/>.
3. Samba File Server : Ubuntu Server Guide. Canonical. URL: <https://ubuntu.com/server/docs/service-samba>.
4. Network File System (NFS) : Ubuntu Server Guide. Canonical. URL: <https://ubuntu.com/server/docs/service-nfs>.
5. Filesystems in the Linux kernel : The Linux Kernel Documentation. Kernel.org. URL: <https://docs.kernel.org/filesystems/>.
6. Hertzog R., Mas R. The Debian Administrator's Handbook. Open Source Edition. URL: <https://debian-handbook.info/>.
7. Shotts W. The Linux Command Line : A Complete Introduction. Internet Edition. No Starch Press. URL: <https://linuxcommand.org/tlcl.php>.
8. Cobbaut P. Linux Networking. Linux Training. URL: <http://linux-training.be/linuxnet.pdf>.
9. Terpstra J. H. Samba-3 by Example : Practical Exercises to Successful Deployment. 2nd ed. Upper Saddle River : Prentice Hall PTR, 2005. URL: <https://www.samba.org/samba/docs/Samba-3-byExample.pdf>.
10. GNU/Linux Advanced Administration. Free Technology Academy. URL: <https://ftacademy.org/materials/fsm/1>.
11. Samba : ArchWiki. Arch Linux. URL: <https://wiki.archlinux.org/title/Samba>.
12. NFS : ArchWiki. Arch Linux. URL: <https://wiki.archlinux.org/title/NFS>.
13. Seifried K. Linux System Administrator's Security Guide. URL: <https://seifried.org/lasg/>.
14. Setting up Samba as a Domain Member : Samba Wiki. URL: https://wiki.samba.org/index.php/Setting_up_Samba_as_a_Domain_Member.

15. Proxmox VE Documentation : офіційна документація. Proxmox Server Solutions GmbH. URL: <https://pve.proxmox.com/pve-docs/>.
16. OpenZFS Documentation. OpenZFS. URL: <https://openzfs.github.io/openzfs-docs/>.
17. Eddy W. Transmission Control Protocol (TCP) : RFC 9293. IETF, 2022. URL: <https://datatracker.ietf.org/doc/html/rfc9293>.
18. SELinux User's and Administrator's Guide. Red Hat. URL: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/using_selinux/.
19. The netfilter.org project. Netfilter Core Team. URL: <https://www.netfilter.org/>.
20. Wireshark User's Guide. Wireshark Foundation. URL: https://www.wireshark.org/docs/wsug_html_chunked/.
21. Securing Debian Manual. The Debian Project. URL: <https://www.debian.org/doc/manuals/securing-debian-manual/>.
22. Carter G., Ts J., Eckstein R. Using Samba : A File and Print Server for Linux, Unix and Mac OS X. 3rd ed. Sebastopol : O'Reilly Media, 2007. 449 p.
23. Eisler M., Labiaga R., Stern H. Managing NFS and NIS. 2nd ed. Sebastopol : O'Reilly Media, 2001. 510 p.
24. Nemeth E., Snyder G., Hein T. R., Whaley B., Mackin D. UNIX and Linux System Administration Handbook. 5th ed. Boston : Addison-Wesley Professional, 2017. 1232 p.

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		98

ДОДАТКИ

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				99
Змн.	Арк.	№ докум.	Підпис	Дата		

Лістинг мережевої конфігурації FileServer

```
ip -br a
sudo nano /etc/netplan/00-installer-config.yaml
network:
version: 2
ethernets:
  enp0s3:
    dhcp4: true
  enp0s8:
    dhcp4: no
addresses:
- 192.168.56.10/24
sudo netplan apply
ip -br a
ping 192.168.56.20
ping 192.168.56.30
systemctl status ssh --no-pager
```

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		100

Лістинг налаштування Samba

```

sudo apt update
sudo apt install -y samba acl
sudo mkdir -p /srv/storage/mixed
sudo mkdir -p /srv/storage/mixed/archive
sudo mkdir -p /srv/storage/mixed/projects
sudo mkdir -p /srv/storage/mixed/public
sudo nano /etc/samba/smb.conf
[mixed]
path = /srv/storage/mixed
browseable = yes
read only = no
valid users = ivan petro
create mask = 0660
directory mask = 0770
sudo testparm
sudo smbpasswd -a ivan
sudo smbpasswd -a petro
sudo systemctl restart smbd
sudo systemctl enable smbd
systemctl status smbd --no-pager
grep -A 15 -n "[mixed]" /etc/samba/smb.conf

```

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		101

Лістинг налаштування NFS

```
sudo apt update
sudo apt install -y nfs-kernel-server
sudo nano /etc/exports
/srv/storage/mixed 192.168.56.0/24(rw,sync,no_subtree_check,root_squash,sec=sys)
sudo exportfs -ra
sudo exportfs -v
sudo systemctl restart nfs-server
sudo systemctl enable nfs-server
systemctl status nfs-server --no-pager
cat /etc/exports
sudo exportfs -v
```

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		102

Лістинг налаштування користувачів, груп і POSIX ACL

```

sudo groupadd fs_data
sudo groupadd fs_rw
sudo groupadd fs_ro
sudo groupadd fs_admins
sudo useradd -m ivan
sudo useradd -m petro
sudo passwd ivan
sudo passwd petro
sudo usermod -aG fs_rw ivan
sudo usermod -aG fs_ro petro
sudo usermod -aG fs_admins administrator
sudo mkdir -p /srv/storage/mixed
sudo mkdir -p /srv/storage/mixed/archive
sudo mkdir -p /srv/storage/mixed/projects
sudo mkdir -p /srv/storage/mixed/public
sudo chown -R root:fs_data /srv/storage/mixed
sudo chmod -R 2770 /srv/storage/mixed
sudo setfacl -m g:fs_rw:rwX /srv/storage/mixed
sudo setfacl -m g:fs_ro:r-x /srv/storage/mixed
sudo setfacl -m g:fs_admins:rwX /srv/storage/mixed
sudo setfacl -m o::--- /srv/storage/mixed
sudo setfacl -d -m g:fs_rw:rwX /srv/storage/mixed
sudo setfacl -d -m g:fs_ro:r-x /srv/storage/mixed
sudo setfacl -d -m g:fs_admins:rwX /srv/storage/mixed
sudo setfacl -d -m o::--- /srv/storage/mixed
getfacl /srv/storage/mixed
id ivan
id petro
ls -la /srv/storage/mixed

```

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		103

Лістинг налаштування UbuntuClient для NFS

```

sudo apt update
sudo apt install -y nfs-common
sudo mkdir -p /mnt/mixed
sudo mount -t nfs 192.168.56.10:/srv/storage/mixed /mnt/mixed
mount | grep mixed
ls -la /mnt/mixed
sudo nano /etc/fstab
192.168.56.10:/srv/storage/mixed /mnt/mixed nfs defaults,_netdev 0 0
sudo systemctl daemon-reload
sudo mount -a
mount | grep mixed
ls -la /mnt/mixed
cat /etc/fstab

```

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		104

Лістинг функціонального тестування SMB/NFS

```

mount | grep mixed
ls -la /mnt/mixed
sudo -u ivan touch /mnt/mixed/nfs_ivan_test.txt
ls -la /mnt/mixed | grep nfs_ivan_test
sudo -u petro touch /mnt/mixed/petro_security_test.txt
sudo -u ivan bash -c 'echo "Line 1 from NFS client" > /mnt/mixed/concurrent_test.txt'
cat /mnt/mixed/concurrent_test.txt
cat /mnt/mixed/concurrent_test.txt
sudo smbstatus --locks
net use Z: | 192.168.56.10\mixed /user:ivan
net use
echo test > Z:\windows_smb_test.txt
dir Z:|
net use Z: /delete
net use Z: | 192.168.56.10\mixed /user:petro
echo test > Z:\petro_windows_test.txt
net use Z: /delete

```

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		105

Лістинг навантажувального тестування

```

iperf3 -s
iperf3 -c 192.168.56.10 -t 30
iperf3 -c 192.168.56.10 -t 30 -R
sudo -u ivan dd if=/dev/zero of=/mnt/mixed/dd_nfs_test.dat bs=1M count=64 conv=fdatasync
status=progress
sudo -u ivan dd if=/mnt/mixed/dd_nfs_test.dat of=/dev/null bs=1M status=progress
fsutil file createnew C:\smb_test_256mb.bin 268435456
robocopy C:\ Z:\ smb_test_256mb.bin /J /NP
mkdir C:\SMBDownload
robocopy Z:\ C:\SMBDownload smb_test_256mb.bin /J /NP

```

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		106

Лістинг тестування безпеки та відмовостійкості

```

sudo apt update
sudo apt install -y nmap
nmap -sV 192.168.56.10
ssh fakeuser@192.168.56.10
sudo journalctl -u ssh --since "10 minutes ago" --no-pager
sudo -u petro touch /mnt/mixed/petro_security_test.txt
sudo -u ivan touch /mnt/mixed/security_root_squash_test.txt
ls -la /mnt/mixed | grep security_root_squash_test
sudo rm -f /mnt/mixed/security_root_squash_test.txt
sudo exportfs -v
sudo reboot
systemctl status ssh --no-pager
systemctl status smbd --no-pager
systemctl status nfs-server --no-pager
mount | grep mixed
ping 192.168.56.10
net use Z: | 192.168.56.10\mixed /user:ivan

```

		Осецький М.П.			ККІКБ.КР.Б. – 123 – 11 - ПЗ	Арк.
		Колощук М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		107