

Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“ ____ ” _____ 2024 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи керування СЗД
на основі SMB Direct”

КБПЗ - 2024

Виконав здобувач вищої освіти
II курсу, групи КІ-23М
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Рудяк А.В.
« ____ » _____ 2024 р.

Керівник проекту
доктор філософії (PhD)
_____ Дреєва Г.М.
« ____ » _____ 2024 р.
Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Рівень вищої освіти магістр
Галузь знань 12 "Інформаційні технології"
Спеціальність 123 "Комп'ютерна інженерія"
Освітньо-професійна (освітньо-наукова) програма "Комп'ютерна інженерія"

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 6 » вересня 2024 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ДРУГИМ (МАГІСТЕРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Рудяку Анатолію Володимировичу

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та програмна реалізація системи керування СЗД на основі SMB Direct

2. Керівник роботи Дреєва Ганна Миколаївна, доктор філософії (PhD)

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 19-13 від 07.08.2024 року

3. Строк подання студентом роботи до захисту 2.12.2024 р.

4. Мета та завдання випускної кваліфікаційної роботи: Метою розробки є дослідження та програмна реалізація системи керування СЗД на основі SMB Direct

5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Призначення та область використання.

6. Наукова новизна.

2. Перегляд аналогічних існуючих систем.

7. Маркетингове та економічне обґрунтування ІТ-проєкту.

3. Опис і обґрунтування проєктних рішень.

8. Заходи з охорони праці та техніки безпеки.

4. Етапи програмування системи.

9. Висновки.

5. Впровадження системи в промислову експлуатацію

6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Наукова новизна

1 аркуш

Структурна схема системи

1 аркуш

Функціональна схема системи

1 аркуш

Діаграма процесів

1 аркуш

Блок-схема алгоритму роботи додатку

2 аркуша

Показники економічної ефективності

1 аркуш

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Доренська А.О.	05.10.2024	14.11.2024
Охорона праці	Марченко К.М., к.т.н., доцент	06.10.2024	16.11.2024

7. Дата видачі завдання « 6 » вересня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.10.2024 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2024 р.	
3.	Розробка моделі компонента	20.10.2024 р.	
4.	Розробка структур даних	25.10.2024 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2024 р.	
6.	Програмування алгоритмів	10.11.2024 р.	
7.	Розрахунок економічної ефективності	13.11.2024 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2024 р.	
9.	Оформлення ПЗ	17.11.2024 р.	
10.	Попередній захист роботи	2.12.2024 р.	

Дата видачі завдання
« 6 » вересня 2024 р.

Підпис керівника

(прізвище та ініціали)Завдання прийнято до виконання
« 6 » вересня 2024 р.

Підпис здобувача

(прізвище та ініціали)

АНОТАЦІЯ

Рудяк А.В. Дослідження та програмна реалізація системи керування СЗД на основі SMB Direct. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2024.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи керування СЗД на основі SMB Direct.

Метою розробки є дослідження та програмна реалізація системи керування СЗД на основі SMB Direct.

Об'єктом дослідження є процес керування СЗД на основі SMB Direct.

Предметом дослідження є методи керування СЗД на основі SMB Direct.

Методи дослідження базуються на методах теорії комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи керування СЗД на основі SMB Direct.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ з ОС Windows 10/11.

Програму розроблено в середовищі Python.

Ключові слова: комп'ютерна інженерія, СЗД, SMB Direct

ABSTRACT

Rudyak A.V. Research and software implementation of the SMB Direct based SMB Direct management system. 123 Computer engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2024.

In this graduation thesis for the second (master's) level of higher education, software is developed, which is intended for the SMB Direct-based SMB Direct management system.

The purpose of the development is the research and software implementation of the SMB Direct-based SMB Direct management system.

The object of the study is the SMB Direct-based SMB Direct management process.

The subject of the study is SMB Direct-based SMB Direct management methods.

Research methods are based on computer network theory methods, mathematical statistics methods, and software development methods.

The result of the work is the software implementation of the SMB Direct based SMB Direct management system.

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

A convenient user interface has been developed. Instructions for working with software tools are provided.

The program can be used on a PC with Windows 10/11 OS.

The program was developed in the Python environment.

Keywords: computer engineering, data storage systems, SMB Direct

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	6
1.1 Призначення системи.....	6
1.2 Область застосування.....	7
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	8
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	8
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	14
2.3 Розгорнута постановка завдання	18
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	20
3.1 Опис функціонування системи	20
3.2 Розробка структурної схеми.....	22
3.3 Розробка функціональної схеми	26
3.4 Розробка діаграми процесів.....	28
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	30
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	30
4.2 Захист розробленого програмного забезпечення.....	45
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	47
6 НАУКОВА НОВИЗНА	52

					ВКРМ-123.24.0037.00.00.ПЗ			
Вим.	Арк.	№ докум.	Підп.	Дата	Дослідження та програмна реалізація системи керування СЗД на основі SMB Direct	Літ.	Аркуш	Аркушіів
Розроб.	Рудяк А.В.					М	1	77
Перев.	Дресва Г.М.							
Н.контр.	Коваленко А.С.					ЦНТУ КІ-23М		
Затв.	Смірнов О.А.							

7	МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ІТ-ПРОЄКТУ	53
7.1	Визначення цільової аудиторії кінцевого готового продукту	53
7.2	Оцінка привабливості шляхом застосування методів експертних оцінок ...	54
7.3	Вибір методу оцінки вартості ПЗ	56
7.4	Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості.....	56
7.5	Пропозиція алгоритму просування проєкту розробки ПЗ	58
7.6	Оптимізація каналів збуту та шляхів реалізації ПЗ	58
7.7	Визначення ключових факторів успіху конкретного проєкту.....	59
8	ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	61
8.1	Вступ.....	61
8.2	Шкідливі і небезпечні фактори при роботі з комп'ютером.....	62
8.3	Аналіз санітарно-гігієнічних умов праці на робочому місці програміста ...	63
8.4	Розробка заходів з умов поліпшення охорони праці	66
8.5	Розрахункова частина	67
9	ОСНОВНІ ВИСНОВКИ.....	69
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	71

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

БД	–	база даних
БЗ	–	база знань
БРС	–	блок розрахункових співвідношень
ВКПК	–	вибір кількості пріоритетних користувачів
ІМА	–	інтелектуальний мережний адміністратор
КОМ	–	корпоративні обчислювальні мережі
ЛОМ	–	локальні обчислювальні мережі
МА	–	мережне адміністрування
ПЗ	–	програмне забезпечення
СЗД	–	системи зберігання даних
СУБД	–	система управління базою даних
ЦОД	–	центр обробки даних
СМІР	–	протокол загальної керуючої інформації
DHCP	–	протокол динамічної конфігурації вузла
FMP	–	HPOpenView Fault Management Platform
MIB	–	база даних інформації керування
OEMF	–	HPOpenView Element Management Framework
OER	–	Optimized Edge Routing
OV	–	HPOpenView
Pf	–	Performance Routing
RMON	–	протокол моніторингу комп'ютерних мереж
SNMP	–	простий протокол керування мережею
WINS	–	служба зіставлення netbios-імен комп'ютерів з ір-адресами вузлів

ВСТУП

Актуальність теми. Триваючий ріст обсягів даних, поява програмно обумовлених систем зберігання, поширення хмарних технологій і флеш-накопичувачів змушують ІТ-спеціалістів адаптувати мережну інфраструктуру СЗД до нових вимог, поліпшувати існуючі й розробляти нові, більше ефективні інтерфейси й протоколи.

Ми живемо в епоху росту обсягів постійно накопичуємих й оброблюваних даних, різних по своїх форматах й іншим характеристикам. Поряд зі зростаючим обсягом корпоративних даних основними драйверами ринку мереж зберігання стають впровадження флеш-накопичувачів і віртуалізація серверів. Щоб можна було ефективно користуватися даними як ресурсом, мати можливість витягати з даних, розміщених на різних носіях і пристроях, корисну інформацію, інтерфейси систем зберігання й мережних комунікацій повинні забезпечувати необхідну продуктивність. Причому доступ до даних повинен бути не тільки швидким, але й недорогим. У світі СЗД технології передачі даних – настільки ж що бурхливо розвивається область, що й технології носіїв, на яких ці дані зберігаються. Цього року очікується подальший ріст швидкостей. Цикл розробки нових стандартів звичайно становить три-чотири роки.

Про що саме мова йде? Про інтерфейси підключення СЗД: Fibre Channel (FC), iSCSI, FCoS, NVMe і NVMe over Fabrics. Про «транспорт»: протоколи передачі даних FC, Ethernet, InfiniBand, SAS. А також про внутрішні інтерфейси систем зберігання, тобто про інтерфейси підключення дисків і флеш-накопичувачів: SATA, SAS і NVMe.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи керування СЗД на основі SMB Direct.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

- Огляд існуючих систем керування СЗД на основі SMB Direct.
- Дослідження системи керування СЗД на основі SMB Direct.
- Програмна реалізація системи керування СЗД на основі SMB Direct.

Об'єктом дослідження є процес керування СЗД на основі SMB Direct.

Предметом дослідження є методи керування СЗД на основі SMB Direct.

Методи дослідження базуються на методах теорії комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

- Удосконалено метод керування СЗД на основі SMB Direct.
- Розроблено вітчизняний продукт керування СЗД на основі SMB Direct,

який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі керування СЗД на основі SMB Direct.

Достовірність наукових результатів підтверджена теоретичними викладеннями, даними комп'ютерного моделювання, коректними дослідженнями параметрів на функціонуючій обчислювальній мережі, а також відповідністю отриманих результатів окремим результатам, наведеним у науковій літературі.

Робота апробована на LVII Науково-технічній конференції здобувачів вищої освіти LV науково-технічної конференції «Наука в ЦНТУ: основні досягнення та перспективи розвитку» (2024 р.), основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №15.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи керування СЗД на основі SMB Direct, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Цього року нас чекає міграція на нові стандарти, під якими буде створюватися відповідна інфраструктура, що включає в себе програмне забезпечення, контролери, мережні інтерфейси й фізичне середовище передачі – кабелі й з'єднувачі. Технологія Fibre Channel (FC) продовжує активно розвиватися. Прийнято стандарт FC Gen6 зі швидкістю передачі даних 32 Гбіт/с, іде робота над його розширенням для підтримки 128 Гбіт/с. Строго говорячи, Fibre Channel Gen6 (шостого покоління) – це FC зі швидкістю передачі 32 і 128 Гбіт/с. Останній варіант ґрунтується на тій же технології FC 32 Гбіт/с і відрізняється лише паралельною конфігурацією (паралельний чотирьохканальний інтерфейс). Комітет T11 привласнив проекту назва FC-PI-6P. В FC 32 Гбіт/с застосовується з'єднувач 25/28G SFP+. В FC 128 Гбіт/с, імовірно, будуть використовуватися з'єднувачі QSFP+, можлива також підтримка з'єднувачів CFP2 або CFP4. Очікується, що перші комутатори Gen6 з'являться в 2017 році. Проект FC-PI-7 припускає розробку стандартів FC із ще більш високими швидкостями – 64 і 256 Гбіт/с. Виразно, у доступному для огляду майбутньому Fibre Channel збереже свої позиції як основна технологія мереж зберігання (SAN), тим більше що в інфраструктуру FC вкладені чималі кошти. Однак згодом у нього можуть з'явитися конкуренти (про це нижче). А от на рівні підключення дисків інтерфейс Fibre Channel може виявитися витиснутим SAS на 12 Гбіт/с (ця технологія дозволяє використовувати всі переваги шини PCIe 3.0).

Тим часом розробляється специфікація інтерфейсу SAS на 24 Гбіт/с: відповідні компоненти можуть з'явитися в 2017 році, а перші продукти – в 2018-му. Як очікується, у фінальних версіях цього інтерфейсу буде використовуватися технологія PCIe 4.x.

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

Відносно FCoS (FC-BB-6) можна почути різні прогнози. Дана технологія для підключення систем зберігання вмерла, так і не злетівши, і цю тему можна закрити. Однак, по оцінках IDC, у найближчі парі років по сукупній ємності підключених зовнішніх СЗД вона буде конкурувати з InfiniBand, але широкого поширення, що очікувалося, все-таки не одержить. FCoS працює поверх Ethernet і може застосовуватися як інтерфейс між мережами FC і Ethernet. Слід зазначити, що як протокол FCoS непопулярний, оскільки не є маршрутизуємим і вимагає застосування спеціальних комутаторів.

1.2 Область застосування

У серпні 2014 року комітет T11 завершив роботу над стандартом FC-BB-6. Використовувана в ньому архітектура VN2VN дозволяє з'єднувати вузли FCoS (Virtual N_Ports) без комутаторів FCoS, що спрощує побудову невеликих мереж. Разом з тим завдяки поліпшеній масштабованості (Domain_ID Scalability) фабрики FCoS можуть обслуговувати великі інфраструктури SAN. Найближча перспектива – перехід на FCoS зі швидкостями 40 і 100 Гбіт/с. Але «злетіти» FCoS навряд чи вже призначено. Інша справа – всюдисущий Ethernet. Стандарти IEEE 802.3ba (Ethernet на 40 і 100 Гбіт/с) минулого ратифіковані в червні 2010 року, що відповідають продукти вже випускаються. Ці технології, а також більше нові версії 25Gb і 50Gb, так само як і 2.5Gb, 5Gb і FC, досить широко висвітлювалися в пресі. Набагато рідше згадується технологія InfiniBand, хоча вона давно й успішно застосовується ряд вендорів і має свої переваги. Через обставини, що відкрилися, на неї варто звернути увагу.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи керування СЗД на основі SMB Direct, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти

Дискові системи й масиви HPE

Дискові платформи D 2000, 3000, 6000 є представниками лінійки JBOD, де представлені полки з наборами накопичувачів. Тут відсутні власні контролери RAID, з'єднання з якими відбувається безпосередньо в сервері або прямо. Концерн HPE здійснює випуск продуктів JBOD, призначених для роботи з дисками різних форматів SSD, SAS, MDL-SAS, обсягу, корпуси й швидкості.

Всі накопичувачі підходять для одночасного монтажу в рамках однієї дискової платформи. Їхня установка або заміна виробляються в гарячому режимі. Також можна міняти й інші елементи начебто вентиляторів або блоків живлення. Доступ до накопичувачів виробляється за допомогою 2-х базових конфігурацій. Одна з них зветься dual domain, позначаючи систему, укомплектовану двома модулями, коли вся передана інформація дублюється, а до кожного накопичувача йде 2 маршрути, що гарантує високий рівень відказостійкості. Друга конфігурація йменується single domain і характеризується простотою.

У дискових платформах D2000, 6000 застосовується інтерфейс 6Gb SAS, а в системах D3000 використовується 12Gb SAS. З його допомогою виробляється підключення до накопичувачів і серверів. Dх600 і D6000 являють собою продукти з функцією підтримки великих дисків, які ставляться до стандарту 3,5. У першому випадку платформа вміщає 1, а в другому 70 дискових накопичувачів.

Розширником масивів P2000 G3, MSA1040 і MSA2040 виступає платформа D2700.

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

Диски малих форматів 2,5 підтримує платформа Dх700, де вони можуть розміщатися в числі 25 шт. Тут мова йде про твердотільні накопичувачі, які підтримує ще й система D3600. Масштабування продуктів Dх600 і Dх700 відбувається за принципом каскаду, максимальне число полиць тут можна довести до 8 шт., коли число великих дисків 3,5 складе 96, а невеликих дисків 2,5 – 200 шт. Застосування контролерів P421, P431, P822 і P841 дозволяє забезпечити змішану експлуатацію моделей Dх600 і Dх700. Число накопичувачів можна довести до 148.

Обидві системи, так само як і рішення D6000, здатні підключатися до блейд-серверів, для чого застосовуються комутатори серії HPE 6Gb SAS BL Switch. З'єднання виробляється за допомогою зонування або виділення з полиці певних накопичувачів конкретному серверу. Отриманий ефект аналогічний прямому з'єднанню з дисками DAS.

Просту технологію підключення великої кількості дискових накопичувачів до розташованого на шасі серверам забезпечує система D6000. Один блейд-сервер здатний обробляти до 6 подібних систем або 420 дискових накопичувачів.

Конфігурація типу dual domain застосовується винятково при роботі з дисками SAS. Коли використовується комутаційна пара 6Gb SAS BL Switch. Разом з тим сервера HPE ProLiant до платформи D6000 можна підключати прямо, використовуючи для конфігурації single domain контролери HPE Smart Array P421, P431, P441, а для конфігурації dual domain контролери P822, P841.

Дискові масиви MSA

В основі лінійки дискових масивів MSA1040 і MSA2040 лежать останні технічні досягнення. У результаті власники можуть створювати на їхній основі ефективні, прості й надійні структури. На сьогодні має місце вже 4 покоління масивів, де є присутнім 1-2 контролери з обсягом пам'яті 6 ГБ кожний (масив MSA2040) або 2 аналогічних контролери (масив MSA1040).

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

Дублювання за допомогою внутрішніх каналів забезпечує підвищену продуктивність, сприяючи й більшій надійності масиву. Також у масивах застосовуються інтерфейси:

- 6/12 Гбіт/с SAS. Вони дозволяють підключати диски;
- Гбіт/с SAS 4x (24 Гбіт/с). Для роботи з полками;
- 8/16 Гбіт/с FC, 6/12 Гбіт/с SAS 4x, 1/10 Гбіт/с iSCSI. Для серверного підключення.



Рисунок 2.1 – Дисккові масиви MSA

У базовій версії обидва масиви випускаються у форм-факторі 2U. Полка вибирається залежно від умов і може вмістити 12 більших або 24 малих диска. При цьому платформа MSA1040 може масштабуватися до 3, платформа MSA2040 до 7 полиць.

При використанні 2-х контролерів відбувається дубляж всіх компонентів, а масив підтримує найбільш затребувані операційні системи. До числа ключових особливостей ставиться застосування конденсаторів, що встигли стати відмінною заміною акумуляторним батареям. Підтримка RAID 0, 1, 3, 5, 6, 10, 50 дозволяє досягти належної надійності й ефективності. Резервні диски, як і інші елементи, можуть мінятися в «гарячому» режимі.

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

До рівня дискових масивів можуть бути оновлені й платформи попереднього покоління MSA P2000, також і масив MSA1040 завжди здатний масштабуватися до MSA2040. Платформи працюють із полками попередніх версій, створюючи два типи копій: моментальний і повний. Знімок відбувається миттєво, що дозволяє завжди відкочуватися до потрібного тимчасового відрізка, зводити нанівець будь-які неполадки. Масив 2040 забезпечує підтримку всіх типів дисків, включаючи SSD, SAS, MDL SAS.

В 2014 році HPE презентувала відновлення системних сховищ, після чого інформація могла оптимізуватися по рівнях, а функціонал віртуалізації істотно розширився. Тепер всі дискові ресурси використовувалися як єдиний кластер, керованість зросла, а час відгуку скоротилося. Практично всі операції стали проходити в автоматичному режимі, позбавивши адміністратора від зайвих турбот.

Опція Thin Provisioning відповідала за виділення системних ресурсів, що надходять у міру необхідності, коли убудований механізм інформував власника про необхідність збільшити число накопичувачів. Під час роботи з комерційними додатками відбувається відновлення ресурсів, за рахунок блоків, що зберігають інформацію.

Ефективність зберігання виросла завдяки опції Thin Reclamation, коли без ручного втручання відбувалося повернення простору, зайнятого вилученими файлами. Оптимізація інформації дає можливість балансування, домагаючись ідеального сполучення вартості, продуктивності й автоматизації.

На зміни вимог основних додатків масиви реагують у реальному часі, а платформа MSA 2040 укомплектована архівною оптимізацією, що коли не використовуються файли автоматично йдуть на дешеві диски.

Оптимальна продуктивність досягається за рахунок Performance Tiering, коли в розпорядженні користувача виявляються всі можливості твердотільних дисків і він без праці працює з найбільш ресурсномісткими додатками. Нарешті обидва масиви дозволяють організувати структуру, стійку до неполадок,

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

перевантаженням і навіть катастрофам, для чого в платформах передбачена функція асинхронної реплікації, з використанням каналів FC- і IP.

IBM Flash System

Стрімкий розвиток інформаційних технологій дозволяє використовувати для ведення бізнесу усе більше нових інформаційних інструментів, які забезпечують високий ріст продуктивності при рішенні критично важливих завдань і обробці масивних потоків різноманітних даних.

Особливістю сучасного бізнесу є постійна конкуренція й боротьба за першість у своїй галузі. Для реалізації цих цілей компанії мають потребу у великому обчислювальному потенціалі. Щоб завжди підтримувати свою діяльність на висоті компанії змушені постійно вкладати величезні фінансові засоби в придбання новітнього встаткування, а також в апгрейд уже наявного.

Зважаючи на те, що інформаційне середовище сучасних бізнес-компаній безупинно міняється, а обсяг використовуваних даних постійно росте, ІТ-інфраструктура вимагає, насамперед, використання інноваційних високопродуктивних систем зберігання даних, які забезпечують високу швидкість доступу до даних, більшу щільність запису, безперебійність роботи й необхідний рівень захисту й безпеки інформації.

Яким образом можна прискорити корпоративні ІТ-системи?

Для збільшення потенціалу свого інформаційного середовища компаніями найбільше часто застосовуються наступні кроки:

– збільшується число робочих ядер центрального процесора – це досить витратний процес, що не завжди відрізняється високою ефективністю стосовно необхідної продуктивності при роботі з даними;

– додаються нові диски в підсистему зберігання даних – досить дорогий процес, що не надає необхідній швидкості відгуку системи зберігання запитуваної інформації;

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

– збільшується обсяг оперативних запам'ятовувальних пристроїв – це рішення не відрізняється високою ефективністю, оскільки вимагає постійного апгрейда, що виражається істотними фінансовими витратами;

– проводиться відновлення використовуваних додатків – досить дорогий спосіб, що вимагає значного часу для реалізації й істотних фінансових витрат.

IBM Flash System – ефективне рішення по прискоренню ІТ-систем від IBM

Компанія IBM постійно працює над створенням нових технологій по роботі з інформаційними архівами, що дозволяє їй пропонувати своїм клієнтам готові рішення, що забезпечують новий рівень продуктивності СЗД.

Останнім з таких рішень компанії є IBM Flash System, що дозволяє одержати необхідний ріст продуктивності, поліпшити ефективність роботи зі СЗД, а також забезпечити високий рівень безпеки в ІТ-середовищу. До основних переваг, які пропонує нове рішення від IBM, можна віднести наступні:

- ріст швидкості роботи критично важливих додатків;
- одержання доступу до необхідної інформації за мінімальні проміжки часу;
- максимальна надійність і відказостійкість;
- простота розгортання, наступного використання й обслуговування.

Застосовуючи IBM Flash System, можна буде вирішити такі часто, що зустрічаються проблеми, як неефективне функціонування серверних вузлів, збої в роботі додатків і затримки в доступі до них, тривале виконання операцій по обробці даних.

Незаперечним фактом є те, що ефективність роботи ІТ-інфраструктури буде тим вище, чим краще функціонує її СЗД. Підтвердженням того, що IBM Flash System істотно впливає на працездатність ІТ-інфраструктури, є наступні статистичні дані. Витрати на придбання програмного забезпечення й підтримки ліцензій скорочуються практично на 50%, швидкість роботи додатків виростає в 6 разів, енерговитрати на живлення й охолодження зменшуються на 75%.

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Python – це об'єктноорієнтована мова програмування високого рівня загального призначення з відкритим кодом. Це визначення може бути важким для новачків, тому розглянемо кожну характеристику окремо, щоб зрозуміти, що вона означає:

- Відкритий вихідний код: це безкоштовно та доступно для подальших покращень, таких як додавання корисних функцій або виправлення помилок.
- Об'єктноорієнтована: заснована не на функціях, але в об'єктах з певними атрибутами й методами.
- Високий рівень: зручний для людини, а не для комп'ютера.
- Загальне призначення: можна використовувати для створення будь-яких програм.

Ця мова використовується в будь-якому програмному забезпеченні, про яке ви тільки можете подумати. Ви можете використовувати його для створення вебсайтів, штучного інтелекту, серверів, програмного забезпечення для бізнесу та багато іншого. Також застосовується в науці про дані, аналізі даних, машинному навчанні, інженерії даних, веброзробці, розробці програмного забезпечення та інших галузях.

Переваги та недоліки Python

Переваги:

– Її легко читати, вчити та писати. Це мова програмування високого рівня з англійським синтаксисом. Це полегшує читання та розуміння коду. Її дійсно легко зрозуміти і вивчити, тому багато людей рекомендують Python новачкам. Вам потрібно менше рядків коду для виконання того ж завдання в порівнянні з іншими основними мовами, такими як C/C++ та Java.

– Підвищує продуктивність. Це дуже продуктивна мова. Завдяки її простоті розробники можуть зосередитися на розв'язанні проблеми. Їм не

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

потрібно витратити багато часу на розуміння синтаксису або поведінку мови програмування. Ви пишете менше коду та виконуєте більше завдань.

– Інтерпретована мова. Python мова, що інтерпретується, а це означає, що вона безпосередньо виконує код по рядку. Якщо сталася помилка, вона зупиняє подальше виконання та повідомляє про її виникнення. Вона показує лише одну помилку, навіть якщо у програмі їх кілька. Це спрощує налагодження.

– Динамічно типізована. Python не визначає тип змінної, доки ми не запустимо код. Вона автоматично надає тип даних, коли відбувається процес виконання. Фахівець може не турбуватися про оголошення змінних та типи даних.

– Безкоштовна та з відкритим вихідним кодом. Ця мова постачається під схваленою OSI ліцензією з відкритим вихідним кодом. Це робить його безкоштовним для використання та розповсюдження. Ви можете завантажити вихідний код, змінити його та навіть розповсюджувати свою версію. Це корисно для організацій, які хочуть використати свою версію для розробки.

– Підтримка великих бібліотек. Стандартна бібліотека Python є величезною, ви можете знайти майже всі функції, необхідні для вашого завдання. Таким чином ви не залежите від зовнішніх бібліотек.

– Портативність. У багатьох мовах, таких як C/C++, потрібно змінити свій код, щоб запустити програму на різних платформах. З Python все інакше. Ви тільки пишете один раз і запускаєте її будь-де.

Недоліки:

– Низька швидкість. Вище ми обговорювали, що це інтерпретована мова з динамічною типізацією. Порядкове виконання коду часто призводить до повільного виконання. Динамічна природа Python також є причиною її низької швидкості, оскільки їй доводиться виконувати додаткову роботу при виконанні коду. Тому вона не підходить для цілей, де швидкість важливий аспект проєкту.

– Неefективна для пам'яті. Ця мова програмування використовує великий обсяг пам'яті, це може бути недоліком при створенні програм, коли віддають перевагу оптимізації пам'яті.

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

– Слабка у мобільних обчисленнях. Python зазвичай використовується у серверному програмуванні. Ми не бачимо – її на стороні клієнта або в мобільних програмах з таких причин: вона не заощаджує пам'ять і має повільну обчислювальну потужність у порівнянні з іншими мовами.

– Доступ до бази даних. Програмувати на цій мові легко, але коли ми взаємодіємо з базою даних, її не вистачає. Рівень доступу до бази даних у Python примітивний та недостатньо розвинений у порівнянні з іншими популярними технологіями.

– Помилки виконання. Це мова з динамічною типізацією, тому тип даних змінної може змінюватись у будь-який час. Змінна, що містить ціле число, у майбутньому може містити рядок, що може призвести до помилок виконання.

Застосування Python:

– Для аналізу даних. Дані стали цінним активом у будь-якій сучасній галузі, і більшість компаній зацікавлені у збиранні, обробці та аналізі релевантних даних, щоб витягти з них цінну інформацію для бізнесу. І тут Python виходить за межі будь-якої конкуренції. Python особливо цінна тим, що крім великої стандартної бібліотеки надає величезний набір додаткових модулів, розроблених спеціально для аналітичних цілей. Найвідоміші бібліотеки Python для аналізу даних – це pandas і NumPy . Ці інструменти дозволяють робити з вашими даними майже все, наприклад, очищати і аналізувати їх, вивчати статистику або візуалізувати приховані тенденції у ваших даних.

– Для візуалізації даних. Візуалізація даних – це окрема частина аналізу даних, яка допомагає нам подавати інформацію, необроблену чи очищену, у більш змістовній формі. Тут Python знову входить у гру, пропонуючи широкий спектр інструментів візуалізації даних. Найпопулярніші з них – matplotlib і заснований на ній seaborn. Використовуючи їх, ми можемо створювати буквально всі види візуалізації: від найпростіших до складніших.

– Для машинного навчання. Машинне навчання (ML) є основою більшості завдань науки даних. Він є областю штучного інтелекту, пов'язаною з використанням алгоритмів, що дозволяють машинам вивчати закономірності та тенденції на основі історичних даних, щоб робити прогнози на основі невідомих

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

даних. – Використовуючи методи ML, ми можемо створювати моделі, які можуть точно передбачити швидкість відтоку клієнтів компанії, оцінити ризик виникнення у людини певного захворювання, визначити оптимальне розташування автомобілів таксі й т.д. За допомогою Python ми можемо побудувати модель ML, використовуючи лише три рядки коду.

– Для розробки програмного забезпечення. Крім свого багатостороннього застосування в галузях науки про дані, Python використовується на кожному етапі розробки програмного забезпечення, включаючи контроль складання, автоматичну безперервну компіляцію, прототипування, відстеження помилок, тестування та обслуговування програмного забезпечення. За допомогою цієї мови можемо створювати аудіо- або відеопрограми на основі методів штучного інтелекту, машинного навчання, API (інтерфейсів прикладного програмування), GUI (графічних інтерфейсів) або будь-якого іншого типу програмного забезпечення.

– Для веброзробки. У той час як для створення візуальної частини вебсайту ми переважно будемо використовувати такі мови, як HTML, CSS та JavaScript, для його невидимої частини ми часто вибираємо Python. Серед масштабних вебсайтів та програм, створених за допомогою цієї мови, варто згадати Google, Facebook, Instagram, YouTube, Dropbox та Reddit.

– Для автоматизації задач/скриптингу. Це відмінний інструмент для написання програм для автоматизації різних завдань, що повторюються. Цей процес називається скриптингом. Зокрема, можна робити скрипти для роботи з файлами та папками. Наприклад, можна створювати, перейменовувати, перетворювати, розділяти, об'єднувати або видаляти файли, перевіряти їх на наявність помилок. Ви також можете використовувати автоматизацію Python для пошуку та завантаження інформації з Інтернету, заповнення та надсилання онлайн-форм та надсилання регулярних повідомлень або електронних листів.

Яким фахівцям потрібно володіти Python:

– Фахівець з даних.

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

- Аналітик даних.
- Інженер даних.
- Інженер з машинного навчання.
- Журналіст даних.
- Архітектор даних.
- Повний стек веброзробника.
- Backend-розробник.
- DevOps-інженер.
- Інженер-програміст

Можемо зробити висновок, що Python ще довго буде популярною мовою, хоч і має низку недоліків. Цю мову використовують для створення вебсайтів, штучного інтелекту, серверів, програмного забезпечення для бізнесу, аналізу даних, машинного навчання, інженерії даних та для багатьох інших областей. Це перспективна і затребувана навичка, яка необхідна у всіх галузях.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи керування СЗД на основі SMB Direct.

В процесі розробки випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформулювати висновки про виконаний обсяг робіт та одержані результати.

КБПЗ - 2024

					VKPM-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

InfiniBand – не настільки популярна технологія, як FC і Ethernet, але вона успішно розвивається, процвітають і компанії, що спеціалізуються в даній області. Показовий приклад – Mellanox, оберт якої росте на 30% щорічно. Компанія випускає ПО, оптичні й мідні кабелі, мережні карти й комутатори InfiniBand від 12 до 648 портів. Останні призначаються для створення обчислювальних кластерів. Наприклад, один з таких кластерів розгорнуть компанією «Т-платформи». Подібні рішення застосовуються у відомій платіжній системі PayPal, на InfiniBand (IB) побудовані системи Oracle Exadata і IBM XIV.

Кожні 2,5 роки Mellanox подвоює швидкість своїх мережних рішень і розширює їхню функціональність. У неї вже є рішення для 100 Гбіт/с (починаючи з кабелів і адаптерів і закінчуючи ПО): адаптери з підтримкою RAID, у тому числі з розподілом RAID по різних вузлах, комутатори IB і Ethernet, мідні й оптичні кабелі.

У продуктах InfiniBand EDR (Enhanced Data Rate) зі швидкістю 100 Гбіт/із застосовуються ті ж з'єднувачі 25/28G SFP+, що й в Ethernet і Fibre Channel. Зараз є технологія InfiniBand High Data Rate (HDR) зі швидкістю 200 Гбіт/с. Імовірно, хост-адаптери HDR будуть розраховані вже на слоти PCIe 4.0.

InfiniBand відома насамперед як технологія кластерного інтерконекта. Її характеризують велика швидкість і низькі затримки, а до недоліків можна віднести високу вартість і складність інфраструктури. У високопродуктивних обчисленнях (HPC) і інших спеціалізованих рішеннях знайшла застосування ще одна технологія, безпосередньо пов'язана з InfiniBand і підтримувана нею. Це Remote Direct Memory Access (RDMA).

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

У 90-і роки проблема із завантаженням процесора при уведенні-виводі була вирішена за допомогою процедури прямого доступу до пам'яті (Direct Memory Access, DMA). Функції роботи з підсистемою уведення-виводу були передані спеціальному контролеру DMA. Однак DMA – це локальні, внутрісистемні операції. Її логічним продовженням стало створення «мережної» версії.

Наприкінці 90-х організація InfiniBand Trade Association запропонувала технологію RDMA. Розроблена для InfiniBand і адаптована для Ethernet, вона дає можливість додатку, що перебуває на одному вузлі, звертатися до ресурсів на іншому вузлі без участі центральних процесорів обох вузлів. За роботу з відправником і одержувачем даних і їхню перевірку в ході операцій відповідає мережний адаптер.

Чому раптом RDMA? З появою флеш-накопичувачів, хмар і програмно обумовлених систем зберігання даних (SDS) ця новинка 1998 року може знайти «другий подих».

Якщо час доступу до HDD характеризується мілісекундами, то для SSD це десятки мікросекунд – в 100 разів швидше. Як очікується, нові технології пам'яті, розроблювальні Intel і Hewlett-Packard Enterprise, збільшать швидкість доступу ще на два порядки. За останнє десятиліття вона виросла приблизно в 10 тис. раз.

Якщо потрібно звертатися до носія через мережу, повинні використовуватися протоколи з мінімальними витратами (overhead). І отут може придатися RDMA.

У свій час через проблеми в реалізації застосування RDMA обмежувалося системами HPC. Однак сьогодні з'явилися нові можливості. За допомогою RDMA можна звертатися до SSD через мережу з тією же швидкістю, що й до локального носія. Адже в RDMA відсутнє багаторазове проміжне копіювання даних при їхній передачі між відправником і одержувачем.

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

Версії RDMA на базі InfiniBand і Ethernet відрізняються методами керування мережею. Для додатків вони прозорі: якщо підтримується одна із цих версій, буде підтримуватися й інша. Різниця – у керуванні мережею.

Що це дає на практиці? Розвантаження процесорів від завдань керування мережею дозволяє створювати гіперконвергентні ЦОДи. Такий ЦОД складається з однакових вузлів – стандартних серверів і СЗД. У середовищі віртуалізації Microsoft Hyper-V за допомогою RDMA можна на порядок збільшити швидкість «живої міграції» віртуальних машин. Все відбувається настільки швидко, що працююче у ВМ додаток цього не помітить. Важливо, що така міграція не вимагає й витрат ресурсів процесорів обох фізичних серверів.

3.2 Розробка структурної схеми

На основі RDMA створений ряд цікавих протоколів і розширень. Наприклад, iSER (iSCSI Extensions for RDMA) – розширення iSCSI для роботи з мереж RDMA. iSCSI дуже популярний і активно розвивається завдяки збільшенню швидкостей Ethernet, але не занадто підходить для завдань, де необхідні малі затримки, до того ж зі збільшенням масштабу систем можуть виникати складності керування. В iSER вся логіка керування й забезпечення надійної передачі даних заснована на стандарті iSCSI, а при операціях читання-запису й передачі даних використовується RDMA.

iSER вважається найефективнішим мережним протоколом SCSI: версія iSCSI із транспортом RDMA працює поверх Ethernet (див. мал. 7) або InfiniBand 10, 25, 40, 50, 56 і 100 Гбіт/с і підтримується всіма стандартними додатками. Це стандарт IETF. Переваги iSER – висока продуктивність (пропускна здатність, IOPS і мала затримка), низьке навантаження на процесор, використання засобів керування iSCSI.

iSER добре підходить для підтримки баз даних, віртуальних середовищ, хмарних сервісів і Web-додатків. Він дозволяє передавати дані в буфер SCSI і з

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

нього, минаючи численні проміжні копіювання. Крім того, вирішується проблема фрагментації TCP, що збільшує затримку через необхідність упорядкування пакетів одержувачем (вони можуть приходити в довільному порядку). В iSER замість звичайного способу інкапсуляції в TCP задіяні можливості RDMA.

Використання логіки керування iSCSI і застосування RDMA в операціях читання й запису дозволяють одержати високу швидкість сервісів даних у віртуальному середовищі. Наприклад, iSER застосовується VMware. Завдяки iSER можна втричі збільшити щільність середовища VDI (кількість VM на одному сервері) у порівнянні з iSCSI over TCP. Протокол iSER підтримується в багатьох ОС. Його використовують і ряд виробників СЗД, зокрема HPE (масиви SL4500), NetApp (E-Series), Violin, Oracle (ZFS).

Ще один цікавий протокол – Microsoft SMB Direct. На відміну від блокового протоколу iSER, це файловий протокол. SMB Direct на базі RDMA дозволяє в п'ять разів підвищити продуктивність у порівнянні з 10Gb. А застосування ще одного розширення – протоколу RoCE (RDMA over Converged Ethernet) з 40Gb в Windows Azure Storage – дало Microsoft можливість заощадити чимало обчислювальних ресурсів.

Як затверджують в Mellanox, мережа RoCE і зменшення навантаження на центральний процесор за рахунок RDMA (RDMA Offload) дозволяють створювати більше ефективні гіперконвергентні ЦОДи й інфраструктури SDS. В останньому випадку ПО SDS забезпечує доступ додатка на одному вузлі до сховища будь-якого іншого вузла. Ресурси зберігання даних різних вузлів утворюють єдиний пул зберігання, доступний додаткам. По такому принципі працює, наприклад, ЦОД Microsoft Azure. Локальні ресурси зберігання вузлів не прив'язані до процесора вузла.

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

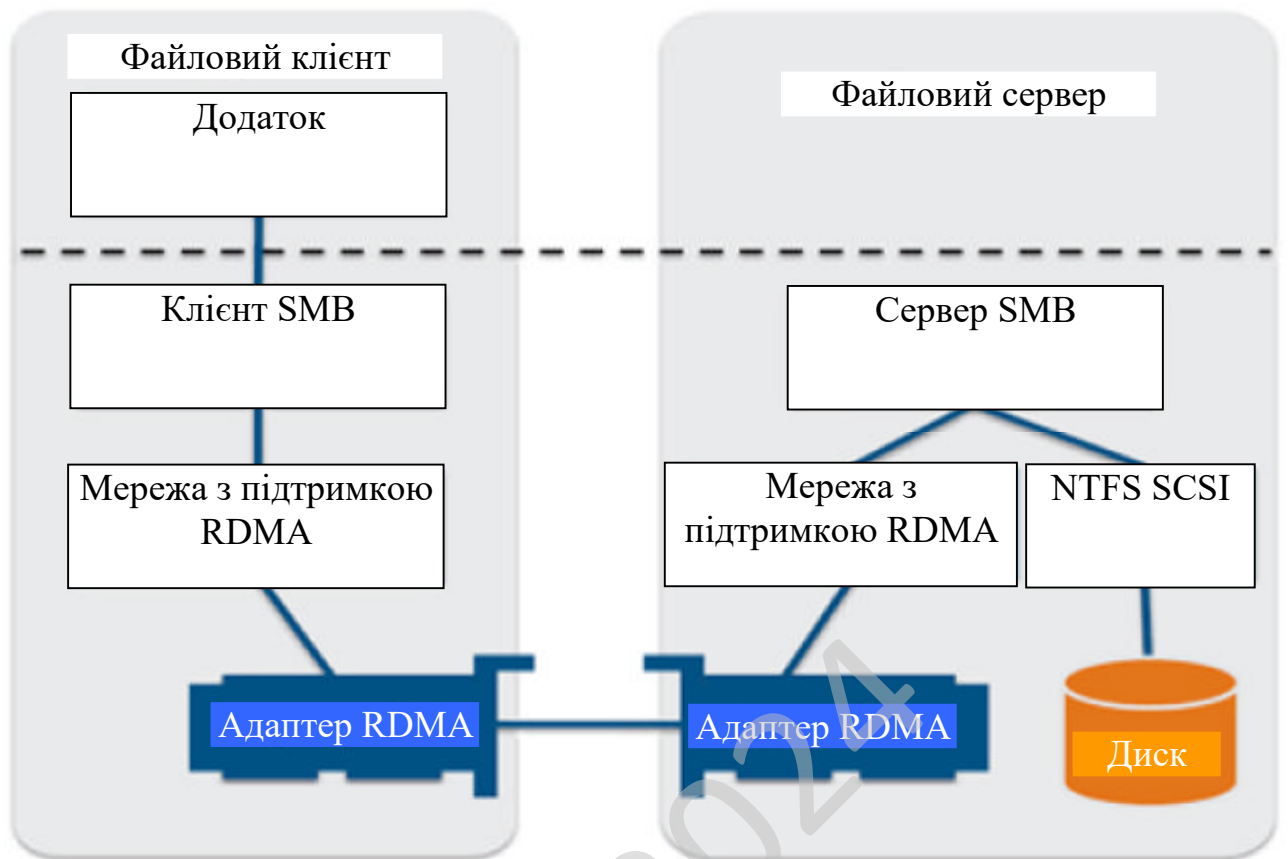


Рисунок 3.1 – Структурна схема системи

За останні 10 років продуктивність процесорів виросла в 10 разів, оперативної пам'яті – в 8 разів, мереж – в 100 разів, внутрішньої шини – в 20 разів, а накопичувачів на жорстких дисках – усього в 1,5 рази. Ситуацію повинна поліпшити флеш-пам'ять. Для роботи із цим носієм розробляються нові протоколи й технології, що розкривають потенціал інтерфейсу PCIe.

Одна з таких технологій – Non-Volatile Memory Express (NVMe). Вона дозволяє усунути значні затримки встаткування й витрати протоколів, пов'язані зі зберіганням, і може застосовуватися для підвищення загальної продуктивності системи, особливо в таких ресурсномістких завданнях, як віртуалізація, Великі Дані й високопродуктивні обчислення. За допомогою NVMe можна одержувати доступ до флеш-пам'яті, минаючи прошарок SCSI, що заощаджує кілька

мікросекунд за часом доступу, особливо у випадку апаратної підтримки NVMe (функції драйвера реалізуються на апаратному рівні).

Масштабована технологія NVMe розроблялася спеціально для прямого підключення пристроїв NVMe (таких як SSD) через PCIe Gen3. Вона забезпечує швидкий доступ процесора до даних у флеш-пам'яті й характеризується, високою продуктивністю в IOPS, малим енергоспоживанням і низькими затримками. Поставки перших продуктів NVMe почалися в 2014 році. В 2016-му багато які вендори серверів представили свої платформи зі слотами NVMe.

Згідно даним Supermicro, у порівнянні з SSD SAS 3.0 сервери з підтримкою NVMe забезпечують у сім разів меншу затримку при доступі до SSD, а пропускна здатність збільшується до шести разів (див. табл. 3). Supermicro позиціонує такі сервери як рішення для завдань HPC, нафтогазової галузі, 3D-Моделювання, графічного дизайну, VDI, віртуалізації, хмарних обчислень і інших ресурсномістких додатків.

NVMe дозволяє підвищити продуктивність системи за рахунок більше повного використання паралелізму пристроїв і програмного забезпечення. Суть технології в тім, що накопичувач стає «ближче» до процесора. Це дає можливість знизити час затримки з 2031 до 86 мкс – більш ніж в 20 разів.

З'явився стандарт NVMe over Fabrics (NVMe). Його завдання – забезпечити ефективний вилучений доступ хосту до пристроїв NVMe через комутуючі фабрики (тобто через мережу), причому майже з такою ж продуктивністю, як і при роботі з локальними пристроями NVMe. Продуктивність створених у цей час прототипів пристроїв становить 450K IOPS при роботі з локальними й вилученими пристроями NVMe PCIe, а при вилученому доступі затримка збільшується незначно.

Стандарт NVMe over Fabrics розроблявся розраховуючи на реалізацію всієї функціональності NVMe поперх Ethernet. Як і NVMe, він підтримує RDMA. NVMe over Fabric може значно поліпшити продуктивність мереж зберігання даних і зараз розглядається як перспективний протокол SAN. У цій області він

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

може потіснити iSCSI, FCoS, FC і InfiniBand, а в перспективі – навіть витиснути традиційний Fibre Channel. Все залежить від того, яку підтримку йому надасть галузь. Можливо подальший розвиток версій NVMe over FC і NVMe over Ethernet. Крім того, він здатний скласти конкуренцію iSER.

У цей час Fibre Channel є де-факто стандартом для побудови великих мереж зберігання даних. Цьому сприяють ряд характеристик, які роблять FC найбільш підходящим протоколом для передачі трафіку систем зберігання: висока швидкість, низькі затримки, механізми контролю пересилання пакетів без втрат, можливість передачі даних на більші відстані. Однак, по даним Dell'Oro Group, за останній рік продажу продуктів FC у світі скоротилися на 3%.

Тим часом QLogic і Brocade продемонстрували в грудні минулого року зразки продуктів NVMe over Fabrics, у яких використовується фабрика Fibre Channel (FC-NVMe). Очікується, що NVMe може стати провідним протоколом для роботи із флеш-масивами.

3.3 Розробка функціональної схеми

Функціональна схема розробленої системи, показана на рисунку 3.2, складається із наступних блоків:

- центрального блоку системи керування СЗД на основі SMB Direct;
- агентів спостереження системи керування СЗД на основі SMB Direct.

Центральний блок системи керування СЗД на основі SMB Direct складається з наступних блоків, які взаємодіють один з одним згідно структурної схеми:

- Планувальник, який визначає які зв'язки потрібно коригувати, або які вузли необхідно внести.
- База даних.
- Блок інтерфейсу ручного вводу, призначений для ручного вводу ймовірного трафіку у тому, або іншому вузлі мережі системи керування СЗД на

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

основі SMB Direct.

– Блок відображення структури та характеристик мережі системи керування СЗД на основі SMB Direct, для наглядного бачення усієї мережі, він може бути виконаний як на логічному так і на фізичному рівні представлення мережі.

– Блок обчислення розрахункових значень, призначений для визначення необхідності зміни топології мережі системи керування СЗД на основі SMB Direct.

– Блок виміру параметрів, призначений для виміру параметрів мережі системи керування СЗД на основі SMB Direct.

– Блок обробки статистики, передачі трафіку у мережі системи керування СЗД на основі SMB Direct.

– Формули розрахунку мережі системи керування СЗД на основі SMB Direct (ФРМ), які дозволяють на основі використання розробленої математичної моделі мережі, з використанням методів та формул теорії масового обслуговування та теорії статистики, розраховувати навантаження трафіку на кожний вузол мережі.

БД використовується для зберігання таблиць, що містять довідкову інформацію й дані, отримані в ході вимірів. Розроблена специфікація моделі й структура бази даних, підтверджують реалізуємість математичної моделі, вбудованої в ІМА, для мережі системи керування СЗД на основі SMB Direct реальної розмірності

Взаємодії блоків інтелектуального адміністратора мережі складається з:

– Бази даних (БД) – блоку, що містить постійно обновлювані дані про мережу.

– Блоку обчислення розрахункових співвідношень (БРС), призначеного для обчислення характеристик мережі.

– Блоку знань (БЗ), що містить формули й характеристики законів розподілів, які буде використовувати для роботи БРС.

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

4 РЕАЛІЗАЦІЯ ПРОЕКТУ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

Блок-схеми є основою ПЗ. Тому від точності і детальності проробки блок-схеми залежить результат всієї програми.

Під час роботи над програмою було створено блок-схеми. Перед їх розглядом необхідно провести роз'яснення який саме тип блок-схем використовується.

Блок-схема це представлення задачі для її аналізу або розв'язування за допомогою спеціальних символів (геометричних образів), які позначають такі елементи, як операції, потік, дані тощо. Блок вхідних та вихідних даних прийнято позначати паралелограмом, блок обчислень (обробки) даних – прямокутником, блок прийняття рішень – ромбом, еліпсом – початок та кінець алгоритму.

У інформаційних технологіях функціональна схема складається з функціональних блоків, які являють собою конструктивно відособлені частини (елементи або пристрої) автоматичних систем, які виконують певні функції. Функціональні блоки на схемі позначають прямокутниками, всередині яких надписують їх найменування відповідно до функцій, що виконуються. Зв'язки між функціональними блоками (внутрішні впливи) позначаються лініями зі стрілками, які вказують напрям впливів.

Функціональні схеми можуть виконуватися в укрупненому і розгорненому вигляді. У першому випадку на схемі зображають найважливіші блоки системи і зв'язки між ними.

У другому варіанті схема відображається більш детально, що полегшує її читання та ілюструє принцип роботи.

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

Основні елементи схем алгоритму це термінатор, процес, рішення, зумовлений процес (підпрограма), дані та з'єднувач.

Термінатор це елемент відображає вхід із зовнішнього середовища або вихід з неї (найчастіше застосування – початок і кінець програми). Всередині фігури записується відповідна дія.

Процес це виконання однієї або кількох операцій, обробка даних будь-якого виду (зміна значення даних, форми подання, розташування). Всередині фігури записують безпосередньо самі операції.

Рішення це показує рішення або функцію перемикального типу з одним входом і двома або більше альтернативними виходами, з яких тільки один може бути обраний після обчислення умов, визначених всередині цього елемента. Вхід в елемент позначається лінією, що входить зазвичай у верхню вершину елемента. Якщо виходів два чи три то зазвичай кожен вихід позначається лінією, що виходить з решти вершин (бічних і нижній). Якщо виходів більше трьох, то їх слід показувати однією лінією, що виходить з вершини (частіше нижній) елемента, яка потім розгалужується. Відповідні результати обчислень можуть записуватися поруч з лініями, що відображають ці шляхи.

Зумовлений процес (підпрограма) це символ відображає виконання процесу, що складається з однієї або кількох операцій, що визначені в іншому місці програми (у підпрограмі, модулі). Всередині символу записується назва процесу і передані в нього дані.

Дані це перетворення у форму, придатну для обробки (введення) або відображення результатів обробки (виведення). Цей символ не визначає носія даних (для вказівки типу носія даних використовуються специфічні символи).

З'єднувач це символ відображає вихід в частину схеми і вхід з іншої частини цієї схеми. Використовується для обриву лінії та продовження її в іншому місці (приклад: поділ блок-схеми, що не поміщається на листі). Відповідні сполучні символи повинні мати одне (при тому унікальне) позначення.

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

При виборі початкової точки відліку при побудові схем було враховано, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об'єктно-орієнтована що вимагає оптимізації, також те, що при розробці програми слід надати особливу увагу модулю системи керування СЗД на основі SMB Direct. Функціональні блоки на схемі позначають прямокутниками, всередині яких надписують їх найменування відповідно до функцій, що виконуються. Зв'язки між функціональними блоками (внутрішні впливи) позначаються лініями зі стрілками, які вказують напрям впливів. Функціональні блоки можуть виконуватися в укрупненому і розгорненому вигляді. У першому випадку на схемі зображають найважливіші блоки системи і зв'язки між ними.

На рисунку 4.1 зображена основна блок-схема програми, на рисунку 4.2 зображено роботу підпрограми.

З яких видно що робота основної програми складається з початкових етапів ініціалізації ПЗ, перевірки наявності ресурсів системи, блоку початку основного циклу з чеканням запиту від користувача в якому відбувається виклик підпрограми та останньої стадії – перевірки поточного стану та поверненням на початок схеми чи з завершенням роботи розробленого ПЗ. При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання. Було використано підходи з використанням UML, це уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування. Є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, називаної UML-моделлю.

UML був створений для визначення, візуалізації, проектування й документування в основному програмних систем. UML не є мовою програмування, але в засобах виконання UML-моделей як інтерпретованого коду можлива кодогенерація.

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

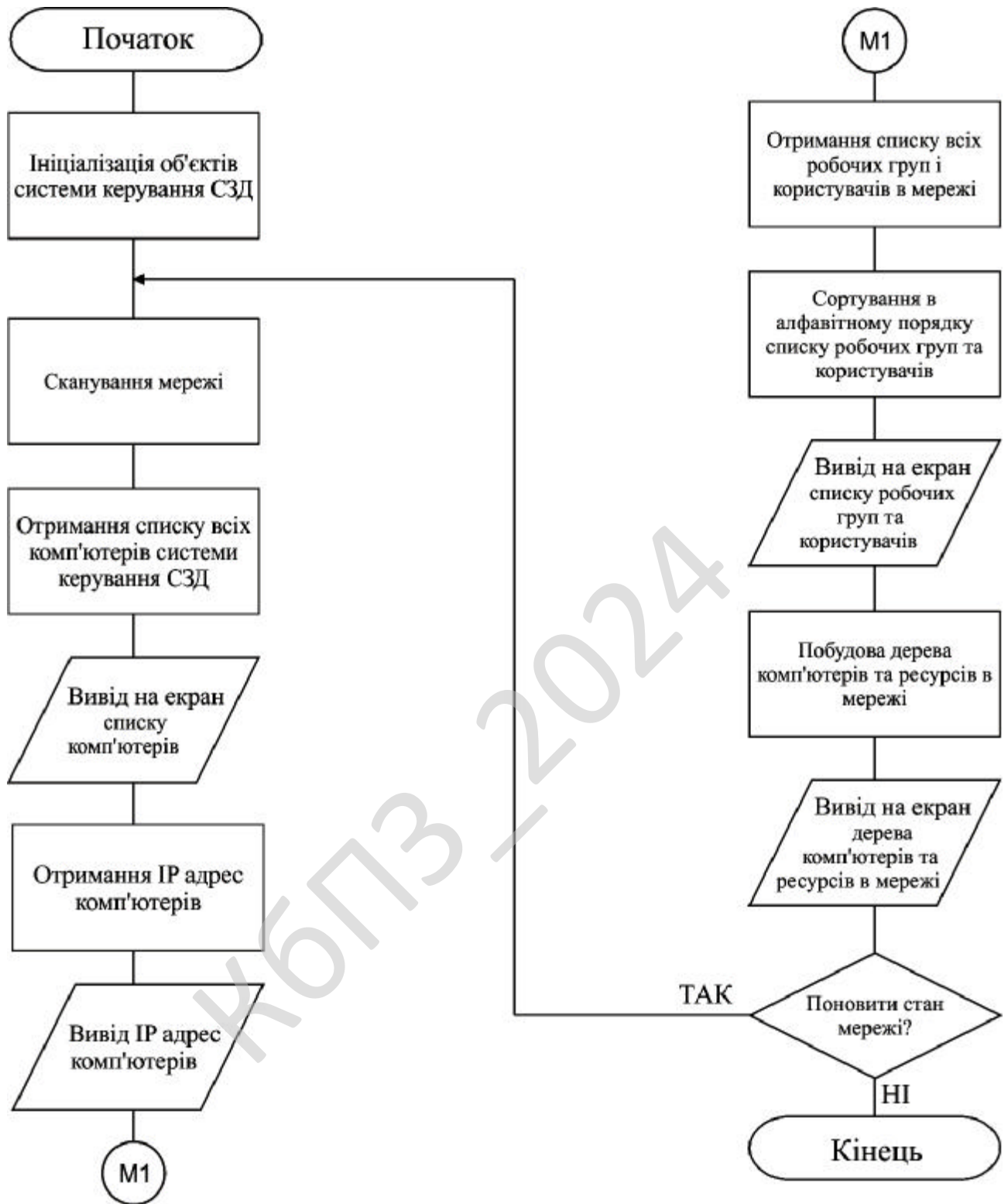


Рисунок 4.2 – Блок-схема роботи підпрограми

Система керування системою збереження даних (СЗД) на основі SMB Direct для випускної кваліфікаційної роботи використовує такі основні компоненти і модулі

1. Модуль аутентифікації користувачів. Він забезпечує захищений доступ до системи і дозволяє виконувати лише авторизованим користувачам операції зі збереження та отримання даних. Для цього використовується модуль os для управління доступами та паролями.

2. Модуль передачі даних через SMB Direct. Використовується протокол SMB Direct для підвищення ефективності передачі даних. Для взаємодії з мережевими функціями задіяно модулі socket та smbprotocol, що дозволяють керувати потоками даних через RDMA (Remote Direct Memory Access).

3. Модуль керування файлами і каталогами. Він надає інтерфейс для операцій з файлами та папками на рівні серверної файлової системи. Використовуються вбудовані модулі os і shutil, що забезпечують переміщення, копіювання та видалення файлів.

4. Модуль логування. Система зберігає всі дії в логах для моніторингу та виявлення можливих проблем. Застосовується модуль logging для відстеження всіх операцій.

Архітектура системи включає:

1. Клієнтський інтерфейс. Користувач взаємодіє із системою через консольний інтерфейс або веб-інтерфейс (за необхідності). Передбачено аутентифікацію, вибір і виконання операцій

2. Сервер передачі даних. Основний сервер приймає запити від клієнтів та керує операціями з файлами через SMB Direct. Він забезпечує безпечну передачу та обробку файлів

3. Модуль збереження. Сервер керує фізичним зберіганням файлів на жорсткому диску або в мережевих сховищах

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35


```

        conn = Connection(uuid=os.urandom(16), server_name=server_ip,
username=username, password=password)
        conn.connect()
        logging.info(f"Connected to SMB server at {server_ip}")
        return conn
    except Exception as e:
        logging.error(f"Failed to connect to SMB server: {str(e)}")
        return None

# Функція передачі файлів через SMB
def transfer_file_via_smb(conn, share_name, local_file, remote_file):
    try:
        tree = TreeConnect(conn, share_name)
        tree.connect()

        # Відкриття віддаленого файлу
        smb_file = Open(tree, remote_file)
        with open(local_file, 'rb') as f:
            data = f.read()
            smb_file.write(data, 0)

        logging.info(f"File {local_file} transferred to {remote_file} via SMB")
    except Exception as e:
        logging.error(f"Failed to transfer file via SMB: {str(e)}")

# Функція для копіювання файлів
def copy_file(source_path, destination_path):
    try:
        shutil.copy(source_path, destination_path)
        logging.info(f"File copied from {source_path} to {destination_path}")
    except Exception as e:
        logging.error(f"Failed to copy file: {str(e)}")

# Функція видалення файлів
def delete_file(file_path):
    try:
        os.remove(file_path)
        logging.info(f"File {file_path} deleted")
    except Exception as e:
        logging.error(f"Failed to delete file: {str(e)}")

# Основна логіка системи

```

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

```

def main():
    # Аутентифікація користувача
    if authenticate_user("admin", "password123"):
        # Підключення до сервера SMB
        smb_conn = connect_to_smb_server("192.168.1.1", "admin", "password123")
        if smb_conn:
            # Передача файлу через SMB
            transfer_file_via_smb(smb_conn, "share", "local_file.txt", "remote_file.txt")
            # Копіювання файлу
            copy_file("local_file.txt", "/backup/local_file.txt")
            # Видалення файлу
            delete_file("local_file.txt")

# Викликаємо основну функцію
if __name__ == "__main__":
    main()

```

Розроблена система забезпечує захищене керування файлами і збереження даних через протокол SMB Direct. Тестування демонструє високу ефективність при великих обсягах даних та високій частоті запитів.

Розглянемо використані технології та їх основні компоненти що підтверджують правильність використаних проектних рішень.

Redmine – вільне серверне ПЗ для управління проектами та відстежування помилок. До системи входить календар-планувальник та діаграми Ганта для візуального представлення ходу робіт за проектом та строків виконання. Redmine написано на мові Ruby і є ПЗ розробленим з використанням відомого веб-фреймворку Ruby on Rails, що означає легкість в розгортанні системи та її адаптації під конкретні вимоги. Для кожного проекту можна вести свої вікі та форуми.

Функціональні можливості:

- Ведення декількох проектів.
- Гнучка система доступу з використанням ролей.
- Система відстеження помилок.
- Діаграми Ганта та календар.
- Ведення новин проекту, документів та управління файлами.

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

- Сповіщення про зміни за допомогою RSS-потоків та електронної пошти.
- Власна Wiki для кожного проекту.
- Форуми для кожного проекту.
- Облік часових витрат.
- Налаштування власних (custom) полів для задач, затрат часу, проектів та користувачів.
- Легка інтеграція із системами керування версіями (SVN, CVS, Git, Mercurial, Bazaar и Darcs).
- Створення записів про помилки на основі отриманих листів
- Підтримка LDAP автентифікації.
- Можливість самореєстрації нових користувачів.
- Багатомовний інтерфейс (у тому числі українська мова).
- Підтримка СКБД: MySQL, PostgreSQL, SQLite.

Діаграма Ганта (Gantt chart, також стрічкова діаграма, графік Ганта) – це популярний тип діаграм, який використовується для ілюстрації плану, графіка робіт за будь-яким проектом. Є одним з методів планування та управління проектами.

Діаграма Ганта являє собою відрізки (графічні плашки), розміщені на горизонтальній шкалі часу. Кожен відрізок відповідає окремому завданню або підзадачі. Завдання і підзадачі, складові плану, розміщуються по вертикалі. Початок, кінець і довжина відрізка на шкалі часу відповідають початку, кінцю і тривалості завдання. На деяких діаграмах Ганта також показується залежність між завданнями.

Діаграма може використовуватися для представлення поточного стану виконання робіт: частина прямокутника, що відповідає завданню, заштриховується, відзначаючи відсоток виконання завдання; показується вертикальна лінія, що відповідає моменту «сьогодні».

Часто діаграма Ганта використовується спільно з таблицею зі списком робіт, рядки якої відповідають окремо взятій задачі, зображеній на діаграмі, а стовпці містять додаткову інформацію про задачу.

Система відстеження помилок Багтрекер – прикладна програма для допомоги розробникам програмного забезпечення (програмістам, тестувальникам тощо) враховувати і контролювати помилки, знайдені у програмах, питання щодо функціональності, рішення та оновлення, побажання користувачів, а також стежити за процесом їх виконання.

Кожному, хто розробляв програмні продукти, добре знайоме співвідношення «20/80» – останні 20 % роботи тривають 80 % часу.

Як це не парадоксально, але нічого дивного в цій пропорції немає, адже саме на завершальній стадії починається тестування проекту, коли виявляються помилки, і що більший проект, то більше буде знайдено помилок.

Водночас досить часто виявляється, що більшість цих помилок були відомі та могли бути виправлені з меншими витратами на попередніх стадіях роботи, але не були вчасно описані, а потім загубилися серед інших важливих завдань.

Отже, система відстеження помилок у найпростішому варіанті – це процес, що включає в себе виявлення помилки, її опис, виправлення і перевірку цього виправлення, тобто процес «стеження» за багом протягом всього як його життєвого циклу, так і життєвого циклу розробки в цілому.

Сукупність інформації про дефект. Головний компонент такої системи – база даних, що містить відомості про виявлені дефекти. Ці відомості можуть включати в себе:

- номер (ідентифікатор) дефекту;
- хто повідомив про дефект;
- дата і час виявлення дефекту;
- версія продукту, в якій виявлено дефект;
- серйозність (критичність) дефекту та пріоритет рішення;
- опис кроків для відтворення дефекту (неправильної поведінки програми);

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

- відповідальний за усунення дефекту;
- обговорення можливих рішень та їх наслідків;
- поточний стан виправлення дефекту;
- версії продукту, в якій дефект виправлений.

Крім того, розвинені системи надають можливість прикріплювати файли, які допомагають описати проблему, наприклад, дампи пам'яті або скріншот.

Використання. Основна перевага систем відстеження помилок полягає в забезпеченні чітких централізованих оглядів, запитів на розробку (включаючи помилки і виправлення) та їх стан. У корпоративному середовищі, системи відстеження помилок можуть бути використані для генерації звітів по продуктивності програмістів виправлення помилок. Однак, це може іноді приводити до неточних результатів, тому що різні помилки можуть мати різні ступені пріоритету та серйозності, що пов'язано з складністю їх фіксації.

Життєвий цикл дефекту. Як правило, система відстеження помилок використовує той чи інший варіант «життєвого циклу» помилки, стадія якого визначається поточним станом помилки.

Типовий життєвий цикл дефекту:

1. Новий – дефект зареєстрований тестувальником.
2. Призначений – призначений відповідальний за виправлення дефекту.
3. Дозволений – дефект переходить назад у сферу відповідальності тестувальника. Як правило, супроводжується резолюцією, наприклад:

- Виправлено (виправлення включені у версію таку-то).
- Дубль (повторює дефект, що вже знаходиться в роботі).
- Не виправлено (працює відповідно до специфікації, має занадто низький пріоритет, виправлення відкладено до наступної версії тощо).
- «В мене все працює» (запит додаткової інформації про умови, в яких дефект проявляється).

4. Далі тестувальник проводить перевірку виправлення, залежно від чого дефект або знову переходить у стан «Призначений» (якщо він описаний як виправлений, але не виправлений), або у стан «Закрито».

5. Відкрито повторно – дефект знайдено знову в іншій версії.

Система може надавати адміністраторові можливість налаштування користувачі, які можуть переглядати і редагувати помилки залежно від їх стану, переводити їх в інший стан або видаляти.

У корпоративному середовищі, система відстеження помилок може використовуватися для отримання звітів, що показують продуктивність програмістів при виправленні помилок. Однак, часто такий підхід не дає достатньо точних результатів через те, що різні помилки мають різну ступінь серйозності та складності. При цьому серйозність проблеми прямо не стосується складності її усунення.

При розробці ПЗ було використано V-Model (або VEE модель) є моделлю розробки інформаційних систем (ИС), спрямованої на спрощення розуміння складнощів, пов'язаних з розробкою систем. Вона використовується для визначення єдиної процедури розробки програмного забезпечення, апаратного забезпечення та людино-машинного інтерфейсу.

Концепція V-подібної моделі була розроблена Німеччиною та США в кінці 1980-х років незалежно один від одного:

– Німецька V-модель була розроблена аерокосмічної компанією IABG в Оттобрунні поряд з Мюнхеном у сприянні з Федеральним департаментом з закупівлі озброєнь в Кобленці, для Міністерства оборони Німеччини. Модель була прийнята німецькою федеральною адміністрацією для цивільних потреб влітку 1992.

– Американська V-Model (VEE) була розроблена національною радою з системної інженерії (міжнародна – з 1995 року) для супутникових систем, включаючи обладнання, програмне забезпечення та взаємодію з користувачами.

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

Сучасною версією V-Model є V-Model XT, яка була затверджена в лютому 2005 року. V-модель використовується для управління процесом розробки програмного забезпечення для німецької федеральної адміністрації.

Зараз вона є стандартом для німецьких урядових і оборонних проектів, а також для виробників ПЗ в Німеччині. V-Model являє собою скоріше набір стандартів у галузі проектів, що стосуються розробки нових продуктів. Ця модель багато в чому схожа з Prince2 і описує методи як для проектного управління, так і для системного розвитку.

Основні принципи

Основний принцип V-подібної моделі полягає в тому, що деталізація проекту зростає при русі зліва направо, одночасно з плином часу, і ні те, ні інше не може повернути назад. Ітерації в проекті виробляються по горизонталі, між лівою і правою сторонами літери. Стосовно до розробки інформаційних систем V-Model – варіація каскадної моделі, в якій завдання розробки йдуть зверху вниз по лівій стороні букви V, а завдання тестування – вгору по правій стороні букви V. У середині V проводяться горизонтальні лінії, що показують, як результати кожної з фаз розробки впливають на розвиток системи тестування на кожній із фаз тестування. Модель базується на тому, що приймально-здавальні випробування ґрунтуються, насамперед, на вимогах, системне тестування – на вимогах та архітектурі, комплексне тестування – на вимогах, архітектурі та інтерфейсах, а компонентне тестування – на вимогах, архітектурі, інтерфейсах та алгоритмах

Цілі

V-модель забезпечує підтримку у плануванні та реалізації проекту. В ході проекту ставляться такі завдання:

1. Мінімізація ризиків: V-подібна модель робить проект більш прозорим і підвищує якість контролю проекту шляхом стандартизації проміжних цілей і опису відповідних їм результатів та відповідальних осіб. Це дозволяє виявляти відхилення в проекті і ризики на ранніх стадіях і покращує якість управління проектом.

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

Переваги:

– У моделі особливе значення надається плануванню, спрямованому на верифікацію та атестацію розроблювального продукту на ранніх стадіях його розробки. Фаза модульного тестування підтверджує правильність деталізованого проектування. Фази інтеграції та тестування реалізують архітектурне проектування або проектування на вищому рівні. Фаза тестування системи підтверджує правильність виконання етапу вимог до продукту і його специфікації.

– У моделі передбачені атестація та верифікація всіх зовнішніх і внутрішніх отриманих даних, а не тільки самого програмного продукту.

– У V-подібної моделі визначення вимог виконується перед розробкою проекту системи, а проектування ПО – перед розробкою компонентів.

– Модель визначає продукти, які повинні бути отримані в результаті процесу розробки, причому кожен отриманий дани повинні піддаватися тестуванню.

– Завдяки моделі менеджери проекту можуть відслідковувати хід процесу розробки, так як в даному випадку цілком можливо скористатися тимчасовою шкалою, а завершення кожної фази є контрольною точкою.

4.2 Захист розробленого програмного забезпечення

Захист розробленого програмного забезпечення буде відбуватися за допомогою алгоритму FEAL – блоковий шифр, запропонований Акіхіро Симідзу і Седзі Міягуті.

У ньому використовуються 64-бітовий блок і 64-бітовий ключ. Його ідея полягає і в тому, щоб створити алгоритм, подібний DES, але з більш сильною функцією етапу. Використовуючи менше етапів, цей алгоритм міг би працювати швидше. На жаль, дійсність виявилася далекою від цілей проекту.

Як вхід процесу шифрування використовується 64-бітовий блок відкритого тексту. Спочатку блок даних підлягає операції XOR з 64 бітами

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

ключа. Потім блок даних розщеплюється на ліву і праву половини. Об'єднання лівої і правої половин за допомогою XOR утворює нову праву половину. Ліва половина і нова права половина проходять через N етапів (спочатку 4). На кожному етапі половина об'єднується за допомогою функції $F[1]$ з 16 бітами ключа і за допомогою XOR – з лівою половиною, створюючи нову праву половину. Вихідна права половина (на початок етапу) стає новою лівою половиною. Після N етапів (ліва і права половини не переставляти після N-го етапу) ліва половина знову об'єднується з допомогою XOR з правою половиною, утворюючи нову праву половину, потім ліва і права об'єднуються разом в 64-бітове ціле. Блок даних об'єднується за допомогою XOR з іншими 64 бітами ключа і алгоритм завершується.

КБПЗ_2024

					VKPM-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розглянемо розроблене ПЗ системи керування СЗД на основі SMB Direct яке зображено на рисунку 5.1. Процес розрахунків здійснюється в консольному додатку з передачею отриманих результатів у користувацький інтерфейс.

З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи:

- Розділу введення даних.
- Вікно виведення результату роботи системи – трафік.
- Вікно ведення журналу роботи системи.
- Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші.
- Функціональних кнопок ПЗ.

Розглянемо процес впровадження програмного забезпечення, це процес налаштування програмного забезпечення під певні умови використання, а також навчання користувачів роботі з програмним продуктом. Впровадження програмного забезпечення це усі дії, що роблять розроблену програмну систему готовою до використання. Даний процес є частинною життєвого циклу програмного забезпечення.

Загалом процес розгортання складається з кількох взаємопов'язаних дій із можливими переходами між ними. Ця активність може відбуватися як з боку виробника так і з боку споживача. Оскільки кожна програмна система є унікальною, то усі процеси та процедури під час розгортання важко передбачити. Тому, "розгортання" можна трактувати як загальний процес відповідно до певних вимог та характеристик. Розгортання може здійснюватись програмістом і в процесі розробки програмного забезпечення.

До діяльностей пов'язаних із розгортанням програмного забезпечення відносять:

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

- Випуск.
- Встановлення та активація.
- Деактивація.
- Адаптація.
- Обновлення.
- Вмонтування.
- Відстежування версій.
- Видалення.
- Вилучення з обігу.

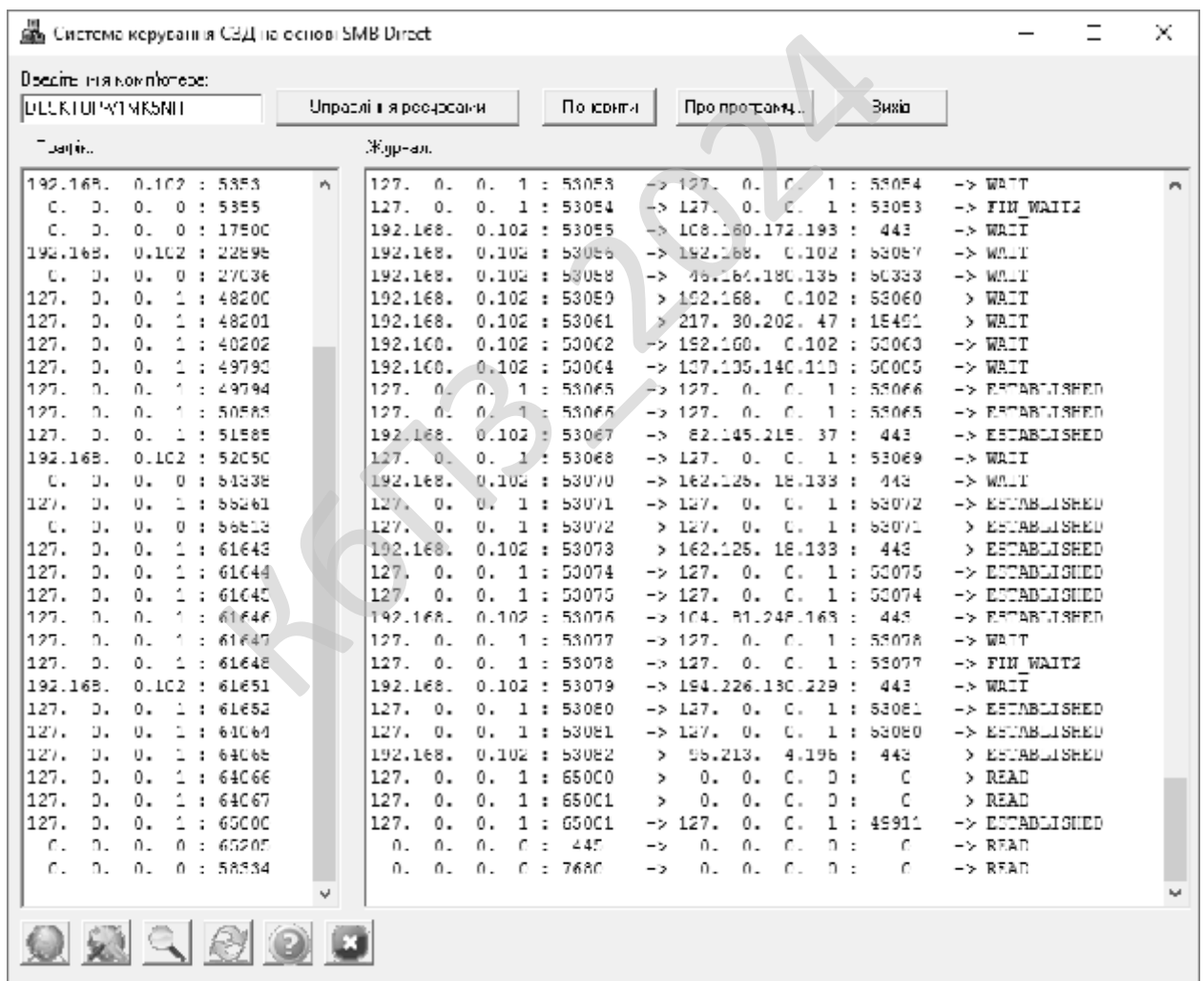


Рисунок 5.1 – Головне вікно ПЗ

При впровадженні програмного забезпечення потрібно урахувати наступні дії:

– Виділення критичних, з точки зору загального результату, процедур в діяльності організації. Коли набір таких процедур визначений, необхідно в першу чергу використовувати ІТ рішення для автоматизації операцій усередині саме цих процедур. Таким чином, розроблене ІТ рішення автоматично стає життєво важливим і затребуваним для організації, а також буде забезпечена публічність процесу впровадження;

– Розширення нормативної бази організації шляхом включення до неї регламентів, що описують порядок виконання процедур автоматизованих процесів. В іншому випадку є небезпека виникнення неузгодженості між автоматизованими процедурами та іншими процесами організації.

– Виконання робіт з загальної стандартизації існуючої діяльності організації, коли виділяються кращі практики виконання процедур і включаються в ІТ рішення за принципом найбільшої корисності для більшості учасників. Відсоток таких процедур щодо загального обсягу автоматизації може бути невеликий, але це надає процесу побудови рішення вагу в організації за рахунок збільшення його необхідності.

Розроблена програма має дуже простий і інтуїтивно зрозумілий інтерфейс з користувачем. Кожен, хто в достатньому обсязі володіє операційним середовищем Windows без особливих складностей освоїть і цю програму, оскільки її інтерфейс інтуїтивно зрозумілий.

Якщо програма не видала ніяких помилок, і працює, то можна використовувати, інакше слід слідувати інструкціям, які пропонує програма.

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

Під час роботи над програмою було проведено тестування програмного забезпечення, тобто технічне дослідження, призначене для виявлення інформації про якість продукту відносно контексту, в якому воно має використовуватись.

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

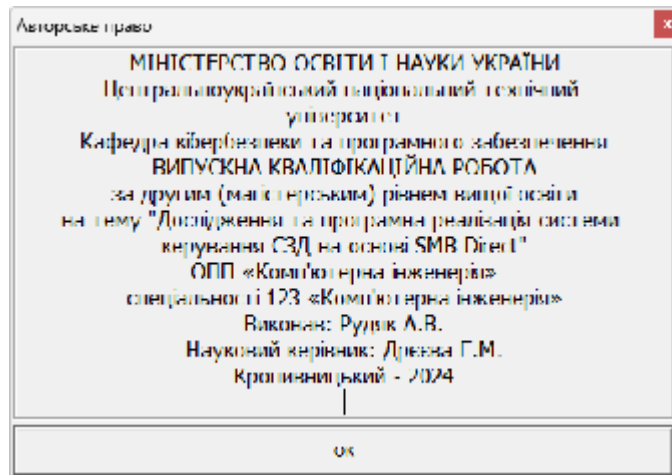


Рисунок 5.2 – Авторське право

Тестування включає як процес пошуку помилок або інших дефектів, так і випробування програмних складових з метою їх оцінки.

Проводилась оцінка:

- відповідності поставленим вимогам;
- правильна відповідь для усіх можливих вхідних даних;
- виконання функцій за прийнятний час;
- практичність;
- сумісність з ОС та стороннім ПЗ.

Оскільки число можливих тестів для програмних компонент практично нескінченне, тому стратегія тестування полягала в тому, щоб провести всі можливі тести з урахуванням наявного часу та ресурсів.

Як результат ПЗ тестувалось стандартним виконанням програми з метою виявлення помилок або інших дефектів.

Обрано умови розповсюдження – commercial software.

Програмне забезпечення, створене комерційною організацією з метою отримання прибутку від його використання іншими, наприклад, шляхом продажу копій.

Найважливішою особливістю комерційних програмних продуктів є підтримка великих компаній, прямо зацікавлених у поширенні програм. Багато

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

організацій надають виключно платну підтримку своїх продуктів, такий підхід, як правило, використовують організації надають відкриті вихідні коди. Для продуктів, що розповсюджуються на комерційній основі діють зазвичай безкоштовні служби підтримки, покликані збільшити рівень довіри у клієнтів і потенційних покупців.

Далеко не завжди, але як правило терміни критично важливих змін в комерційних продуктах значно менше, ніж у некомерційних проектів.

Це пов'язано з тим, що над комерційним продуктом працюють цілі групи розробників і ця робота є їх основним заняттям. Розробникам-початківцям як правило доводиться шукати додаткові способи заробітку, і це збільшує час, що витрачається на доповнення і зміни програм.

Так як основним рушійним фактором створення комерційного ПЗ є одержання прибутку, то комерційні програмні продукти першими заповнюють вільні ніші та пропонують варіанти вирішення завдань відразу по мірі виявлення вакууму в будь-якому секторі ринку.

Окремий вид комерційних програм, коли їх розробка оплачується безпосередньо замовником. Такі програми найчастіше позбавлені всіх переваг комерційних продуктів, оскільки мають обмежений бюджет, але більш адаптовані до вимог замовника, ніж аналоги.

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи керування СЗД на основі SMB Direct.

Метою розробки є дослідження та програмна реалізація системи керування СЗД на основі SMB Direct.

Об'єктом дослідження є процес керування СЗД на основі SMB Direct.

Предметом дослідження є методи керування СЗД на основі SMB Direct.

Методи дослідження базуються на методах теорії комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

- Удосконалено метод керування СЗД на основі SMB Direct.
- Розроблено вітчизняний продукт керування СЗД на основі SMB Direct, який має більш широкі можливості, на відміну від існуючих аналогів.

					VKPM-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

7 МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ІТ-ПРОЄКТУ

7.1 Визначення цільової аудиторії кінцевого готового продукту

Результати дослідження та програмної реалізації системи керування системами зберігання даних (СЗД) на основі SMB Direct можуть бути цікаві різним групам осіб та організацій (рисунок 7.1).

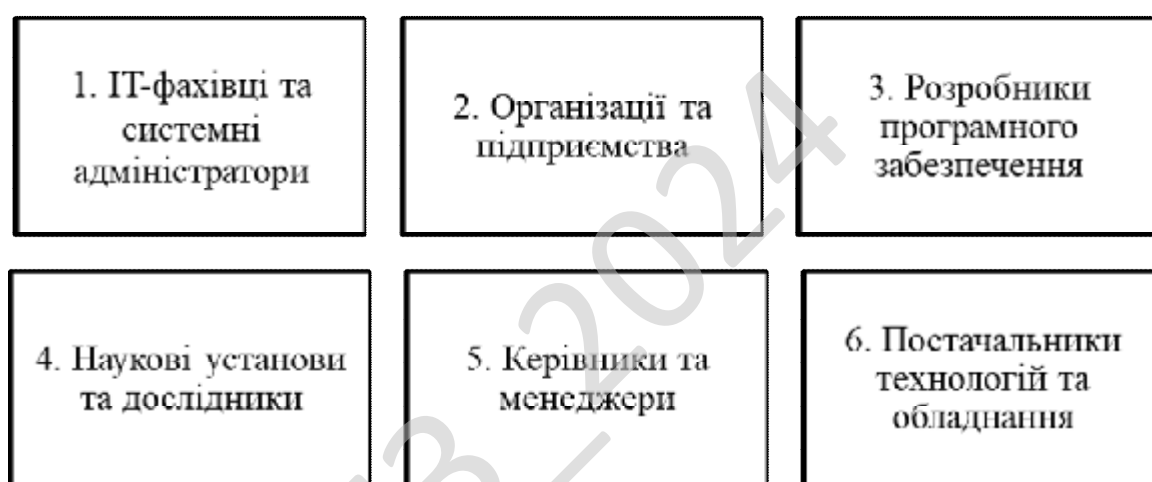


Рисунок 7.1 – Цільова аудиторія

Можемо надати охарактеризувати цільову аудиторію наступним чином. Системні адміністратори, які займаються управлінням інфраструктури зберігання даних, можуть знайти корисними результати, що стосуються оптимізації продуктивності та масштабованості СЗД. Інженери з безпеки зацікавлені в методах забезпечення безпеки даних та захисту інформації під час передачі через SMB Direct.

Великі підприємства, які мають значні обсяги даних і потребують ефективних рішень для управління та зберігання даних. Хостинг-провайдери та дата-центри зацікавлені для застосування результатів для покращення своїх послуг з зберігання даних і забезпечення надійності.

Розробники систем керування можуть використовувати результати для вдосконалення своїх продуктів, інтеграції з SMB Direct та підвищення продуктивності своїх рішень. Розробники мережевих технологій зацікавлені у впровадженні нових функцій, що поліпшують передачу даних через мережі.

Академічні дослідники можуть використовувати результати для подальших наукових досліджень у галузі комп'ютерних наук, зберігання даних та мережевих технологій. Фахівці, які надають консультації щодо технологічних рішень у сфері зберігання даних та мереж.

ІТ-менеджери зацікавлені у стратегічних рішеннях для оптимізації витрат та підвищення продуктивності в організації. Фінансові аналітики можуть бути зацікавлені у розрахунках рентабельності та економічної ефективності впровадження системи.

Виробники обладнання, які виготовляють апаратуру для зберігання даних, можуть бути зацікавлені у нових можливостях для інтеграції своїх продуктів з SMB Direct. Компанії, що надають програмні та апаратні рішення для управління даними, можуть використовувати результати для вдосконалення своїх продуктів.

Результати дослідження та програмної реалізації системи керування СЗД на основі SMB Direct мають потенціал залучити широкий спектр зацікавлених сторін, включаючи ІТ-фахівців, підприємства, розробників, науковців та менеджерів, які прагнуть оптимізувати процеси управління даними та зберігання даних у своїх організаціях.

7.2 Оцінка привабливості шляхом застосування методів експертних оцінок

Оцінка привабливості програмної реалізації системи керування СЗД на основі SMB Direct може бути проведена шляхом застосування методів експертних оцінок.

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

7.3 Вибір методу оцінки вартості ПЗ

Вибір методу оцінки вартості для програмної реалізації системи керування СЗД на основі SMB Direct залежить від специфіки проекту, наявності ресурсів, цілей оцінки та даних, які можна використовувати. Для програмної реалізації системи керування СЗД на основі SMB Direct рекомендується комбінувати кілька методів. Наприклад, ви можете розпочати з методу порівняння витрат для отримання початкової оцінки, а потім використовувати детальний розрахунок для точнішої оцінки. Використання експертних оцінок може також допомогти врахувати специфічні аспекти, які можуть бути пропущені в інших методах. Оптимальний підхід до оцінки вартості залежить від конкретних обставин проекту, тому важливо враховувати наявні дані, ресурси та цілі оцінки при виборі методу.

7.4 Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості

Економічна ефективність впровадження системи керування системами зберігання даних (СЗД) на основі SMB Direct може бути оцінена через різні аспекти, такі як зниження витрат, підвищення продуктивності, поліпшення доступності та покращення управлінських процесів.

Для оцінки економічної ефективності важливо визначити основні показники, які будуть використовуватися в розрахунках: витрати на обладнання та програмне забезпечення: вартість впровадження системи (сервери, програмне забезпечення для управління СЗД); операційні витрати: витрати на електроенергію, охолодження, технічне обслуговування та підтримку; витрати на адміністрування: час і ресурси, витрачені на управління системою; продуктивність: час обробки запитів, зниження затримок, збільшення пропускну здатності.

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

7.5 Пропозиція алгоритму просування проекту розробки ПЗ

Пропонуємо алгоритм просування проекту програмної реалізації системи керування СЗД на основі SMB Direct, який включає основні етапи та дії для забезпечення успішної реалізації та впровадження (рисунок 7.2).

Цей алгоритм просування проекту програмної реалізації системи керування СЗД на основі SMB Direct охоплює всі ключові етапи, які допоможуть забезпечити успішну реалізацію проекту, залучення клієнтів та підвищення їх задоволеності. Важливо постійно адаптуватися до змін на ринку та потреб клієнтів для досягнення максимальної ефективності.

7.6 Оптимізація каналів збуту та шляхів реалізації ПЗ

Оптимізація каналів збуту та шляхів реалізації проекту програмної реалізації системи керування СЗД на основі SMB Direct є критично важливими для досягнення успіху та максимізації доходів. Ось кілька стратегій та рекомендацій, які можуть допомогти у цьому процесі:

- визначення цільових каналів збуту;
- створення партнерської екосистеми;
- використання цифрового маркетингу;
- запровадження програми лояльності;
- забезпечення підтримки та навчання;
- використання аналітики для оптимізації;
- розробка нових продуктів та послуг.

Оптимізація каналів збуту та шляхів реалізації проекту програмної реалізації системи керування СЗД на основі SMB Direct вимагає комплексного підходу. Важливо враховувати потреби клієнтів, розвивати партнерські відносини, використовувати цифрові технології та постійно вдосконалюватися на основі аналітики. Це дозволить забезпечити успішну реалізацію проекту та підвищити рівень задоволеності клієнтів.

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

1. Аналіз ринку та цільової аудиторії

Дослідження ринку: Вивчіть тенденції в галузі управління СЗД, конкурентів та потреби споживачів.
Визначення цільової аудиторії: Визначте основних користувачів (підприємства, організації, IT-відділи) і їхні потреби.

2. Розробка унікальної торговельної пропозиції (УТП)

Визначення переваг: Окресліть основні переваги вашої системи (швидкість, безпека, сумісність, зниження витрат).
Формулювання УТП: Створіть чітке повідомлення про те, чому ваша система краще, ніж у конкурентів.

3. Створення маркетингової стратегії

Вибір каналів просування: Визначте найбільш ефективні канали (соціальні мережі, контент-маркетинг, email-розсилки, вебінари).
Розробка контенту: Створіть матеріали (статті, відео, інфографіку), які демонструють переваги та особливості вашої системи.
Планування рекламних кампаній: Визначте бюджет та формати реклами для різних каналів.

4. Проведення демонстрацій та вебінарів

Організація вебінарів: Проведіть безкоштовні вебінари, щоб продемонструвати функціональність системи та її переваги.
Індивідуальні демонстрації: Запропонуйте потенційним клієнтам безкоштовні консультації та демонстрації системи.

5. Налагодження партнерств

Співпраця з IT-компаніями: Укладіть партнерські угоди з компаніями, які можуть інтегрувати вашу систему у свої рішення.
Співпраця з консультантами: Заточайте експертів для популяризації системи серед клієнтів.

6. Збір та аналіз відгуків

Опитування користувачів: Проводьте опитування серед раних користувачів для отримання відгуків про систему.
Аналіз результатів: Використовуйте отримані дані для покращення системи та маркетингової стратегії.

7. Оптимізація та вдосконалення

Аналіз кампаній: Оцінюйте ефективність маркетингових кампаній за допомогою аналітичних інструментів.
Внесення змін: На основі аналізу вносьте корективи в стратегію просування, щоб покращити результати.

8. Підтримка та обслуговування

Створення служби підтримки: Запропонуйте користувачам доступ до технічної підтримки та документів.
Регулярні оновлення: Запровадьте регулярні оновлення програмного забезпечення для підвищення функціональності та безпеки.

Рисунок 7.2 – Алгоритм просування проекту

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

7.7 Визначення ключових факторів успіху конкретного проєкту

Ключові фактори успіху проєкту програмної реалізації системи керування Системами Зберігання Даних (СЗД) на основі SMB Direct можуть варіюватися, проте наведені аспекти на рисунку 7.3 є основними.



Рисунок 7.5 – Ключові фактори успіху проєкту

Успіх проєкту програмної реалізації системи керування СЗД на основі SMB Direct залежить від комбінації технічних, маркетингових, фінансових та організаційних факторів. Систематичний підхід до всіх цих аспектів допоможе досягти цілей проєкту та забезпечити його довгостроковий успіх на ринку.

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Наявний в даний час в нашій країні комплекс розроблених організаційних заходів та технічних засобів захисту, накопичений передовий досвід роботи ряду обчислювальних центрів показує, що є можливість домогтися значно більших успіхів у справі усунення впливу на працюючих небезпечних і шкідливих виробничих факторів. Проте стан умов праці та його безпеки в ряді обчислювальних центрів (ОЦ) та підприємств ще не задовольняють сучасним вимогам. Оператори ЕОМ, оператори підготовки даних, програмісти та інші працівники ОЦ та підприємств ще стикаються з впливом таких фізично небезпечних і шкідливих виробничих факторів, як підвищений рівень шуму, підвищена температура зовнішнього середовища, відсутність або недостатня освітленість робочої зони, електричний струм, статична електрика і інші.

Багато працівників ОЦ та підприємств пов'язані з впливом таких психофізичних факторів, як розумова перенапруга, перенапруження зорових і слухових аналізаторів, монотонність праці, емоційні перевантаження. Вплив зазначених несприятливих факторів призводить до зниження працездатності, викликане розвиваються втому. Поява і розвиток втоми пов'язане зі змінами, які виникають під час роботи в центральній нервовій системі, з гальмівними процесами в корі головного мозку. Наприклад сильний шум викликає труднощі з розпізнаванням колірних сигналів, знижує швидкість сприйняття кольору, гостроту зору, зорову адаптацію, порушує сприйняття візуальної інформації, зменшує на 5 – 12% продуктивність праці. Тривала дія шуму з рівнем звукового тиску 90 дБ знижує продуктивність праці на 30 – 60%.

Медичні обстеження працівників ОЦ та підприємств показали, що крім зниження продуктивності праці високі рівні шуму призводять до погіршення

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

слуху. Тривале перебування людини в зоні комбінованого впливу різних несприятливих факторів може призвести до професійного захворювання. Аналіз травматизму серед працівників ВЦ показує, що в основному нещасні випадки відбуваються від впливу фізично небезпечних виробничих факторів при заправці носія інформації на обертний барабан при знятому кожусі, під час співробітниками невластивих їм робіт. На другому місці випадки, пов'язані з дією електричного струму.

8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

Електронно-обчислювальна машина (ЕОМ) та інше обладнання є джерелами небезпеки ураження електричним струмом. Так як робота програміста характеризується істотним зоровим навантаженням, то вимагає належного освітлення. У приміщенні, в якому працюють люди (у т.ч. програмісти) необхідно створити належний мікроклімат, параметри якого регламентуються, Державними санітарними правилами і нормами, зокрема ДСанПіН 3.3.2.007-98.

При роботі з використанням ЕОМ відзначають наступні небезпечні та шкідливі фактори:

- ризик виникнення надзвичайних ситуацій природного або штучного характеру на об'єкті або території.
- ризик виникнення пожежі;
- негативний вплив на органи зору людини;
- ризики ураження електричним струмом;
- недостатня, або надмірна освітленість робочого місця;
- електромагнітні (у т.ч. високочастотні) випромінювання (коливання);
- несприятливі мікрокліматичні умови;
- нервово-емоційна напруженість праці;
- інтелектуальні навантаження;
- монотонність праці;

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

обчислювальних машин» [5], але відповідають нормативним вимогам Наказу Міністерства соціальної політики України № 207, від 14.02.2018 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» [5] та НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин»). Таним чином можна зробити висновок, що санітарно-гігієнічні умови праці на робочому місці програміста відповідають вимогам.

Температура повітря в приміщенні визначається впливом температури зовнішнього повітря і тепловою енергією, яка виділяється всередині приміщення. Джерелами виділення теплоти в даному приміщенні є електроустаткування, освітлювальні прилади, а також люди. У світлий час доби джерелом надлишкового тепла є сонячна радіація. Згідно Постанови № 42 від 01.12.1999 Головного державного санітарного лікаря України, робота, виконувана в даному приміщенні, відноситься до категорії Іа. В цьому випадку людина витрачає енергії до 120 ккал у годину. Вологість повітря в приміщенні визначається впливом багатьох факторів, серед яких: вологість атмосферного повітря, виділення вологи людьми (при диханні та випарами з поверхні шкіри).

Мікроклімат повітряного середовища в приміщенні характеризується запиленістю та загазованістю повітря. Мікроклімат приміщення визначається діючим на організм людини поєднанням, вологості, температури, швидкості руху повітря та інтенсивності теплового випромінювання. Аналіз мікроклімату складається з визначення зазначених вище факторів і порівняння результатів із встановленими нормами. У таблиці 8.3 наведено оптимальні та фактичні значення параметрів мікроклімату як для категорії ваги робіт Іа, так і розглянутого приміщення. У приміщеннях, де встановлено ЕОМ, рекомендується застосування тільки оптимальних значень показників мікроклімату.

Проведений аналіз показує, що показники мікроклімату в приміщенні відповідають установленим нормам. Штучне опалення застосовується у холодний період року.

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

світло на ділянці периферійного зору значно збільшує напруженість очей і, як наслідок, призводить до їх швидкої стомлюваності, ступінь освітлення приміщення і яскравість екрану комп'ютера повинні бути приблизно однаковими.

8.4 Розробка заходів з умов поліпшення охорони праці

Згідно аналізу умов праці в розглянутому приміщенні, ми одержали наступні результати:

- розмірі приміщення, у розрахунку на одному працюючого, відповідають нормативам;
- мікроклімат відповідає нормативному значенню;
- акустичні умови роботи не перевищують нормативних значень;

Таким чином можна припустити, що основною причиною можливого зниження працездатності програміста є психофізіологічний фактор, тому основна пропозиція буде така: дотримання позитивної психологічної атмосфери в колективі та регламентованого режиму праці та відпочинку, організація робочого місця з урахуванням ергономічних вимог.

Рекомендовані заходи: регулярні періодичні наочні огляди персоналом шляхів для евакуації людей із приміщення, відповідно до плану евакуації (який повинен розташовуватись на видному місці у приміщенні), включення до колективного договору мінімально можливого вмісту аптечок з обов'язковою наявністю масок-клапанів, або іншого спорядження для штучного дихання. Регулярна періодична перевірка параметрів заземлення та занулення (вимірювання опору ланцюга).

Регулярна наочне знайомство персоналу із шляхами для евакуації людей із приміщення відповідно до плану евакуації, забезпечення розподільних щитів спеціальними розетками з заземлюючими контактами; організація заземлення всіх приладів і пристроїв, які працюють при напрузі вище 36 В.

Так як при ураженні електричним струмом у людини може статися фібриляція шлуночків серця, в організації бажано мати дефібрилятор і підготовлений персонал для роботи з ним.

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

8.5 Розрахункова частина

Проведемо розрахунок штучного освітлення за методом коефіцієнта використання світлового потоку для приміщення ширина якого складає 5 м, довжина – 5,95 м, висота – 2,8 м.

У зазначеному приміщенні працює 4 людей.

Для того, щоб визначити потрібну кількість світильників, які повинні забезпечити нормований рівень освітленості, визначимо світловий потік, що падає на робочу поверхню за формулою [1]:

$$F = E \cdot S \cdot K \cdot Z / n,$$

де:

F – світловий потік, що розраховується, Лм;

E – нормована мінімальна освітленість, Лк; $E = 300$ Лк;

S – площа освітлюваного приміщення (у нашому випадку $S = 5 \times 5,95 = 29,7$ м²);

K – коефіцієнт запасу, що враховує зменшення світлового потоку лампи в результаті забруднення світильників в процесі експлуатації (його значення залежить від типу приміщення і характеру робіт, що проводяться в ньому, в нашому випадку $K = 1,5$);

Z – відношення середньої освітленості до мінімальної (зазвичай приймається рівним 1.1... 1.2, в нашому випадку $Z = 1,1$);

n – коефіцієнт використання світлового потоку, (відношення світлового потоку, що падає на розрахункову поверхню, до сумарного потоку всіх ламп і обчислюється в долях одиниці; залежить від характеристик світильника, розмірів приміщення, забарвлення стін і стелі, що характеризуються коефіцієнтами відбиття від стін ($\rho_{стін}$) і стелі ($\rho_{стелі}$), значення коефіцієнтів дорівнюють $\rho_{стін} = 50\%$ і $\rho_{стелі} = 50\%$.

Обчислимо індекс приміщення за формулою:

$$i = S / (h \cdot (A + B)),$$

де:

S – площа приміщення, $S = 29,7$ м²;

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

h – розрахункова висота підвісу, $h = 3$ м (співпадає з висотою стелі, т.я. лампи освітлення закріплюються на стелі);

A – ширина приміщення, $A = 5$ м;

B – довжина приміщення, $B = 5,95$ м.

Підставимо всі значення у формулу та визначимо індекса приміщення:
 $i=1,4$.

Знаючи індекс приміщення, за знаходимо $n = 0,29$ (з табличних даних коефіцієнтів використання світлового потоку (n) світильників з відповідним типом лампам) [8]. Підставимо всі значення у формулу, визначимо світловий потік: $F=64027$ Лм.

Для розрахунку думемо використовувати світлодіодні стельові панелі *Призма-72 6400K*, світловий потік яких $F_{\text{л}} = 7200$ Лм.

Число ламп визначається по формулі:

$$N=F/F_{\text{л}}$$

де: F – світловий потік,

$F_{\text{л}}$ – світловий потік однієї лампи.

Підставимо всі значення у формулу та визначимо індекса приміщення:

$$N= 64027 / 7200=8,8 \text{ шт.}$$

Приймаємо необхідну кількість світлодіодних світильників 9 шт.

Висновки до розділу

Дотримання всіх необхідних умов праці не лише сприяє збереженню здоров'я працівників, а також підвищує ефективність виробництва в цілому.

З цих міркувань було здійснено аналіз умов праці, призначеного для праці програмістів, проведено розгляд небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи. Виконано розрахунок штучного освітлення, як одного з ключових факторів впливу на працездатність та здоров'я програміста. Розроблено заходи з умов поліпшення охорони праці.

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи керування СЗД на основі SMB Direct.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти наведені теоретичне узагальнення й рішення наукового завдання дослідження методів керування СЗД на основі SMB Direct.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем керування СЗД на основі SMB Direct.

– Досліджена система керування СЗД на основі SMB Direct.

– На основі отриманих результатів досліджень створена програмна реалізація системи керування СЗД на основі SMB Direct.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання керування СЗД на основі SMB Direct.

Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

При створенні програмного забезпечення було використано об'єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

Програма реалізована на мові високого рівня Python. Дана мова програмування дозволяє найбільш ефективно обробляти дані. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows 10/11.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм FEAL.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

Проведено маркетингове та економічне обґрунтування ІТ-проєкту, що дозволило визначити ключові фактори успіху даного проєкту.

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Рудяк А.В. Дослідження та програмна реалізація системи керування СЗД на основі SMB Direct // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2024.
2. Ramon Nastase «Computer Networking: The Beginner’s guide for Mastering Computer Networking, the Internet and the OSI Model». 2018. – 186 p.
3. Russ White & Ethan Banks «Computer Networking Problems and Solutions: An Innovative Approach to Building Resilient, Modern Networks». 2017. – 832 p.
4. Kuznetsov, O., Kryvinska, N., Ilchenko, O., Smirnova, T., Ulianovska, Y. «Comparative Analysis of Cryptocurrency Trading Platforms Using the Analytic Hierarchy Process». *CEUR Workshop Proceedings*, 2023, 3628, pp. 106-115.
5. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56.
6. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yenchев, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings*, Volume 3530, 2023, pp. 256-265.
7. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». *Lecture Notes on Data Engineering and Communications Technologies*, 2023, 178, pp. 208–223.
8. Smirnova, T., Gnatyuk, S., Yudin, O., Sydorenko, V., Polozhentsev, A., «The Model for Calculating the Quantitative Criteria for Assessing the Security Level of Information and Telecommunication Systems». *CEUR Workshop Proceedings Volume 3156*, 2022, Pages 390-399.

					БКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

9. Smirnova T., Gnatyuk S., Berdibayev R., Avkurova Zh., Iavich M. «Cloud-Based Cyber Incidents Response System and Software Tools». *Communications in Computer and Information Science*, 2021, vol 1486. Springer, Cham. pp 169-184.

10. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». *CEUR Workshop Proceedings*. Volume 2740, 2020, Pages 102-114.

11. Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». *Journal of theoretical and applied information technology* Vol.98. No 21, 2020, P. 3334-3346.

12. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Ukraine, Kyiv, May 14-18. 2020. P. 172-177.

13. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) *Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies*, vol 48. Springer, Cham. 2021. pp 557-587.

14. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». *CEUR Workshop Proceedings* Volume 2616, 2020, Pages 125-136.

15. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». *CEUR Workshop Proceedings* Volume 2616, 2020, Pages 366-379.

16. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated

					БКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

with the Markov chains», *CEUR Workshop Proceedings* Volume 2608, 2020, Pages 633-645.

17. Smirnov O., Kuznetsov A., Zaichenko Yu., Pastukhov M., Oleshko O., Kuznetsova K., «Formation of Discrete Signals with Special Correlation Properties». *International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2019*; Odessa; Ukraine; 9-13 September 2019. P.22-28.

18. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». *International Journal of Computing*; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.

19. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*, Kyiv, Ukraine, 8 – 11 October 2019 . P.517-522.

20. Smirnov, O., Odarchenko, R., Abakumova, A., Usik, P., Kundyz, M., «QoE optimization technique for media delivery in 5G networks». *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*, Kyiv, Ukraine, 8 – 11 October 2019. P.597-601.

21. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». *CEUR Workshop Proceedings*, Vol 2588, P. 90-106, 2019.

22. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Averchev, A., Pastukhov, M., Kuznetsova, K., «Formation of Pseudorandom Sequences with Special Correlation Properties», *2019 3rd International Conference on Advanced Information and Communications Technologies, AICT -2019/ Lviv, Ukraine, 2-6 July, 2019*, P. 395-399.

23. Smirnov, O., Kuznetsov, A., Kiian, A., Zamula, A., Rudenko, S., Hryhorenko, V., «Variance Analysis of Networks Traffic for Intrusion Detection in

					БКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

Smart Grids», 2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS), Kyiv, Ukraine April 17-19, 2019 P. 353-358.

24. Smirnov, O., Kuznetsov, A., Kavun, S., Babenko, B., Nakisko, O., Kuznetsova, K., «Malware Correlation Monitoring in Computer Networks of Promising Smart Grids», 2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS), Kyiv, Ukraine April 17-19, 2019 P. 347-352.

25. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special Correlation Properties», CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 618-629.

26. Smirnov A.A., Kuznetsov A.A., Danilenko D.A., Berezovsky A., «The statistical analysis of a network traffic for the intrusion detection and prevention systems», *Telecommunications and Radio Engineering*. – Volume 74, Issue 1. – Begel House Inc. – 2015. – P. 61-78.

27. Батрак О., Смірнова Т., Гнатюк В., Одарченко Р., Смірнов О. «Дослідження показників ефективності функціонування та перспектив розвитку систем IP-телефонії». *Підводні технології*, 2024, № 13, с. 28-35.

28. Аль-Мудхафар Акіл Абдулхуссейн М., Смірнова Т.В., Буравченко К.О., Смірнов О.А. «Метод оцінки та підвищення користувальницького досвіду абонентів в програмно-конфігурованих мережах на основі використання машинного навчання». *Сучасні інформаційні системи*, 2023, том 7, № 2, С. 49-56.

29. Смірнова Т.В., Гнатюк С.О., Сидоренко В.М., Юдін О.Ю., Сидоренко С.Ю., «Модель визначення критичності галузевих інформаційно-телекомунікаційних систем». *Проблеми інформатизації та управління*, № 2(70). 2022. С. 28-37.

30. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Смірнов С.А., Поліщук Л.І., «Дослідження стійкості до диференціального криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 3(69). С. 93-98.

31. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Поліщук Л.І., Смірнов С.А. «Дослідження статистичної стійкості та швидкісних характеристик запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Вісник Хмельницького національного університету. Серія: «Технічні науки»*, № 2 (307). С. 46-52. 2022.

32. Смірнов О.А., Смірнова Т.В., Константинова Л.В., Смірнов С.А., Якименко Н.М., «Дослідження стійкості до лінійного криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 1(67). С. 84-89.

33. Смірнов О.А., Смірнова Т.В., Буравченко К.О., Кравченко С.С., Горбов В.О., «Хмарна система підтримки прийняття рішень технологічного процесу відновлення поверхонь конструкцій і деталей машин». *Сучасні інформаційні системи*. 2021. Т. 5, № 4. С. 79-95

34. Смірнов О.А., Усік П.С., Миронець І.В., Буравченко К.О., Якименко Н.М. «Метод підвищення ефективності розподіленої обробки даних у комп'ютерних системах операторів стільникового зв'язку» *Вісник Черкаського державного технологічного університету. Технічні науки*. №4. С. 103-110. 2020.

35. О.А.Смірнов, Т.В.Смірнова, Л.І. Поліщук, К.О. Буравченко, А.О.Макевнін, «Дослідження хмарних технологій як сервісів», *Кібербезпека: освіта, наука, техніка*. № 3(7). С. 43-62. 2020.

36. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В., Поліщук Л.І. Інформаційна безпека в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2020. – 294 с.

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

37. О.А. Смірнов, П.С. Усік, «Дослідження перспектив використання технологічних рішень в мережах 5G» у *Кібербезпека та інформаційні технології: монографія*. – Х. : ТОВ «ДІСА ПЛЮС», 2020.С. 122-135.

38. Смірнов О.А., Дреєва Г.М., Дреєв О.М., Смірнова Т.В. «Фрактальний аналіз генератора самоподібного трафіку на основі ланцюга Маркова». *Центральноукраїнський науковий вісник. Технічні науки*. № 2(33). с. 161-172, 2019.

39. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В. Поліщук Л.І. Проектування комп'ютерних систем та мереж. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2019. – 264 с.

40. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).

41. Смірнов О.А., Дреєва Г.М. Метод генерування фрактального трафіку за допомогою моделі генератора на графі. Монографія: Інформаційна безпека та інформаційні технології : монографія / за заг. ред. В. С. Пономаренка. – Х. : Вид. Рожко С.Г. 2019. С. 123-139

42. Дреєва Г.М., Смірнов О.А., Дреєв О.М. Метод генерування фрактальноподібної числової послідовності на основі скінченного автомату для моделювання трафіку у мережі. *Центральноукраїнський науковий вісник. Технічні науки*. № 1(32). с. 173-183, 2019.

43. Смірнова Т.В., Солових Є.К., Смірнов О.А., Дреєв О.М. Побудова хмарних інформаційних технологій оптимізації технологічного процесу відновлення та зміцнення поверхонь деталей. *Центральноукраїнський науковий вісник. Технічні науки*. № 1(32). с. 184-194, 2019.

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

44. Смірнов О.А., Смірнов С.А., Поліщук Л.І., Смірнова Т.В., Коноплицька-Слободенюк О.К. Метод формування антивірусного захисту даних з використанням безпечної маршрутизації метаданих. Кібербезпека: освіта, наука, техніка. – Том 3 № 3. – Київ: КУ ім. Бориса Грінченка. – 2019. – С. 63-87.

45. Смірнов О.А., Гнатюк С.О., Кавун С.В., Терейковський І.А., Жмурко Т.О., Смірнов С.А., Коваленко А.С. Основи безпеки в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2018. – 177 с.

46. Смірнов О.А., Котелянець В.В. Стійкі до колізій стохастичні моделі функціонування безпроводових сенсорних мереж. Вісник інженерної академії України, №3, с. 145-152, 2018

47. Смірнов О.А., Смірнов С.А., Дідик А.К., Дреєв А.М. Алгоритми формування безлічі маршрутів передачі метаданих у антивірусні хмарні системи. Збірник наукових праць "Системи обробки інформації". – Випуск 5 (142). – Х.: ХУПС – 2016. – С. 148-152.

48. Смірнов О.А., Смірнов С.А., Дідик А.К., Дреєв О.М. Моделі системи нейромережових експертів безпечної маршрутизації у хмарних антивірусних системах. Збірник наукових праць "Системи обробки інформації". – Випуск 3 (140). – Х.: ХУПС – 2016. – С. 36-39.

49. Смірнов О.А., Смірнов С.А., Дідик А.К., Дреєв А.М. Спосіб контролю ліній зв'язку телекомунікаційної системи антивірусу. Збірник наукових праць Харківського університету Повітряних Сил. Випуск 2 (47). – Харків: ХУПС. – 2016. – С. 121-127.

50. Смірнов О.А., Смірнов С.А., Дідик А.К. Метод безпечної маршрутизації метаданих у хмарні антивірусні системи. Системи озброєння та військова техніка. – Випуск 2 (46) – Х.: ХУПС – 2016. – С. 146-149.

51. Смірнов О.А., Кавун С.В., Доренський О.П., Вялкова В.І. Інформаційна безпека в комп'ютерних мережах. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 151 с.

					ВКРМ-123.24.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

Додаток А
(обов'язковий)

Технічне завдання

Зміст

1 Найменування та область застосування.....	2
2 Підстава для розробки.....	2
3 Мета та призначення розробки.....	2
4 Джерела розробки.....	2
5 Технічні вимоги.....	2
5.1 Вміст проекту.....	2
5.2 Показники призначення.....	3
5.3 Вимоги до функціональних характеристик.....	3
5.4 Вимоги до архітектури.....	3
5.5 Вимоги до надійності.....	3
5.6 Умови експлуатації.....	4
5.7 Вимоги до складу та параметрів технічних засобів.....	4
5.8 Вимоги до інформаційної і програмної сумісності.....	4
5.8.1 Обладнання.....	4
5.8.2 Мова програмування.....	4
5.8.3 Вхідні дані.....	5
5.8.4 Вихідні дані.....	5
6 Вимоги до програмної документації.....	5
7 Економічні вимоги.....	5
8 Вимоги щодо охорони праці.....	5
9 Перелік документів, що розробляються.....	6
10 Етапи розробки.....	6
11 Порядок контролю та приймання.....	6

					ВКРМ-123.24.0037.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Рудяк А.В.				<i>Дослідження та програмна реалізація системи керування СЗД на основі SMB Direct</i>	Літ.	Аркуш	Аркушів
Перевірів	Дресва Г.М.					М	1	6
Н. Контр.	Коваленко А.С.				ЦНТУ КІ-23М			
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи керування СЗД на основі SMB Direct.

2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 19-13 від 07.08.2024 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи керування СЗД на основі SMB Direct.

4 Джерела розробки

Джерелом цієї випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є стосовна до теми література і існуючі аналоги.

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;
- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;

					ВКРМ-123.24.0037.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

- техніко-економічне обґрунтування доцільності прийнятого до розробки програмного забезпечення;
- аналіз умов праці;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

- програмну реалізацію системи керування СЗД на основі SMB Direct;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно мати обмежень на версію драйверів та операційної системи.

5.4 Вимоги до архітектури

Компонент, що розробляється повинен використовувати системні засоби та апаратні засоби, що на даному етапі розвитку обчислювальної техніки найбільше поширені.

5.5 Вимоги до надійності

Програмні модулі написані по всім правилам, які стосуються стандартних викликів процедур, функцій, методів і форм, визначених технічною документацією на середовище розробки.

					ВКРМ-123.24.0037.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

- температура повітря: 19-20 град. по Цельсію;
- відносна вологість повітря до 80%;
- атмосферний тиск 107 кПа.

5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ, працювати в ОС Windows 10/11 і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Windows 10/11.

5.8.1 Обладнання

Комп'ютер Intel® Celeron/8 Mb/1.2 Gb/SVGA 14" 1Mb або сумісні з ним.

5.8.2 Мова програмування

Середовище Python.

					ВКРМ-123.24.0037.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у виді опису структури даних, схем та опису алгоритму, а також текстів вихідних модулів програмного забезпечення згідно ЄСПД .

7 Економічні вимоги

7.1 Для ПЗ необхідно виробити функціонально-вартісний аналіз варіантів розробки.

7.2 Виконати маркетингове та економічне обґрунтування ІТ-проєкту з урахуванням цін на 3 вересня 2024 року.

8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинні бути розглянуті шкідливі і небезпечні фактори при роботі з комп'ютером.

					ВКРМ-123.24.0037.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

9 Перелік документів, що розробляються

- Наукова новизна – 1 аркуш.
- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Показники економічної ефективності – 1 аркуш.
- Пояснювальна записка – 77 аркушів.

10 Етапи розробки

10.1 Збір і обробка інформації по темі випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти. Постановка задачі на виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти (складання ТЗ).

10.2 Проведення досліджень або експериментальних робіт для уточнення основних положень випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.

10.3 Розробка функціональних схем, блок схем алгоритмів роботи програмного забезпечення.

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Маркетингове та економічне обґрунтування ІТ-проєкту.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

11.1 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на попередній захист 02.12.2024 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 16.12.2024 р.

					ВКРМ-123.24.0037.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

Керівник випускної кваліфікаційної роботи за
другим (магістерським) рівнем вищої освіти

_____ Дресва Г.М.

*Дослідження та програмна реалізація
системи керування СЗД на основі SMB Direct*

Лістинг програми

Код документу 18

Носій: CD/DVD-диск / USB-флеш-накопичувач

Загальна кількість аркушів: 22

Літера: РП

Кропивницький – 2024 року

Основна програма

```
# Імпортуємо необхідні модулі
import os
import shutil
import socket
import logging
from smbprotocol.connection import Connection
from smbprotocol.open import Open
from smbprotocol.tree import TreeConnect
from smbprotocol.file_info import FileAttributes, FileBasicInformation
from smbprotocol.file_info import FileStandardInformation

# Налаштовуємо логування
logging.basicConfig(filename='system_smb_logs.log', level=logging.INFO)

# Функція для створення каталогу
def create_directory(directory_path):
    try:
        if not os.path.exists(directory_path):
            os.makedirs(directory_path)
            logging.info(f"Directory {directory_path} created")
        else:
            logging.info(f"Directory {directory_path} already exists")
    except Exception as e:
        logging.error(f"Failed to create directory: {str(e)}")

# Функція для копіювання файлу між локальними каталогами
def local_copy_file(source_file, destination_file):
    try:
        shutil.copy(source_file, destination_file)
        logging.info(f"File copied from {source_file} to {destination_file}")
    except Exception as e:
        logging.error(f"Failed to copy file: {str(e)}")

# Функція для видалення файлу
def delete_local_file(file_path):
    try:
        if os.path.exists(file_path):
            os.remove(file_path)
            logging.info(f"File {file_path} deleted")
        else:
            logging.info(f"File {file_path} does not exist")
    except Exception as e:
        logging.error(f"Failed to delete file: {str(e)}")

# Функція для перевірки розміру файлу
def check_file_size(file_path):
    try:
        file_size = os.path.getsize(file_path)
        logging.info(f"File size of {file_path}: {file_size} bytes")
        return file_size
```

```

except Exception as e:
    logging.error(f"Failed to check file size: {str(e)}")
    return None

# Функція для перевірки існування файлу
def file_exists(file_path):
    exists = os.path.exists(file_path)
    if exists:
        logging.info(f"File {file_path} exists")
    else:
        logging.info(f"File {file_path} does not exist")
    return exists

# Функція для аутентифікації користувача
def authenticate_user(username, password):
    logging.info(f"Authenticating user {username}")
    if username == "admin" and password == "password123":
        logging.info(f"User {username} authenticated successfully")
        return True
    else:
        logging.info(f"User {username} authentication failed")
        return False

# Функція для підключення до сервера SMB
def connect_to_smb_server(server_ip, username, password):
    try:
        conn = Connection(uuid=os.urandom(16), server_name=server_ip,
username=username, password=password)
        conn.connect()
        logging.info(f"Connected to SMB server at {server_ip}")
        return conn
    except Exception as e:
        logging.error(f"Failed to connect to SMB server: {str(e)}")
        return None

# Функція для створення дерева з'єднання
def smb_tree_connect(conn, share_name):
    try:
        tree = TreeConnect(conn, share_name)
        tree.connect()
        logging.info(f"Connected to share {share_name}")
        return tree
    except Exception as e:
        logging.error(f"Failed to connect to share {share_name}: {str(e)}")
        return None

# Функція для передачі файлу через SMB
def transfer_file_via_smb(tree, local_file, remote_file):
    try:
        with open(local_file, 'rb') as f:
            data = f.read()

        smb_file = Open(tree, remote_file)

```

```

smb_file.create(FileAttributes.FILE_ATTRIBUTE_NORMAL)
smb_file.write(data, 0)
smb_file.close()

logging.info(f"File {local_file} transferred to {remote_file} via SMB")
except Exception as e:
    logging.error(f"Failed to transfer file via SMB: {str(e)}")

# Функція для отримання інформації про файл на SMB сервері
def get_file_info(tree, remote_file):
    try:
        smb_file = Open(tree, remote_file)
        smb_file.create(FileAttributes.FILE_ATTRIBUTE_NORMAL)

# Отримуємо базову інформацію про файл
        file_info = smb_file.query_info(FileBasicInformation)
        logging.info(f"File {remote_file} basic info: {file_info}")

# Отримуємо стандартну інформацію про файл
        standard_info = smb_file.query_info(FileStandardInformation)
        logging.info(f"File {remote_file} standard info: {standard_info}")

        smb_file.close()
    except Exception as e:
        logging.error(f"Failed to get file info for {remote_file}: {str(e)}")

# Функція для видалення файлу на SMB сервері
def delete_file_via_smb(tree, remote_file):
    try:
        smb_file = Open(tree, remote_file)
        smb_file.create(FileAttributes.FILE_ATTRIBUTE_NORMAL)
        smb_file.delete()
        smb_file.close()

        logging.info(f"File {remote_file} deleted from SMB server")
    except Exception as e:
        logging.error(f"Failed to delete file via SMB: {str(e)}")

# Функція для перевірки доступу до файлу на SMB сервері
def check_access_via_smb(tree, remote_file):
    try:
        smb_file = Open(tree, remote_file)
        smb_file.create(FileAttributes.FILE_ATTRIBUTE_NORMAL)
        access_granted = smb_file.check_access(0x00120089) # Read and write
access
        logging.info(f"Access to {remote_file} checked: {access_granted}")
        smb_file.close()
        return access_granted
    except Exception as e:
        logging.error(f"Failed to check access to {remote_file}: {str(e)}")
        return False

# Функція для запису даних у файл на SMB сервері

```

```
def write_data_to_smb(tree, remote_file, data):
    try:
        smb_file = Open(tree, remote_file)
        smb_file.create(FileAttributes.FILE_ATTRIBUTE_NORMAL)
        smb_file.write(data.encode(), 0)
        smb_file.close()

        logging.info(f"Data written to {remote_file} on SMB server")
    except Exception as e:
        logging.error(f"Failed to write data to {remote_file}: {str(e)}")

# Функція для читання даних з файлу на SMB сервері
def read_data_from_smb(tree, remote_file):
    try:
        smb_file = Open(tree, remote_file)
        smb_file.create(FileAttributes.FILE_ATTRIBUTE_NORMAL)

        data = smb_file.read(1024, 0)
        smb_file.close()

        logging.info(f"Data read from {remote_file} on SMB server")
        return data.decode()
    except Exception as e:
        logging.error(f"Failed to read data from {remote_file}: {str(e)}")
        return None

# Основна функція системи
def main():
    # Аутентифікація користувача
    if authenticate_user("admin", "password123"):
        smb_conn = connect_to_smb_server("192.168.1.1", "admin", "password123")

        if smb_conn:
            tree = smb_tree_connect(smb_conn, "share")

            if tree:
                # Передача файлу через SMB
                transfer_file_via_smb(tree, "local_file.txt", "remote_file.txt")

                # Перевірка існування файлу
                file_exists("local_file.txt")

                # Копіювання файлу на локальному рівні
                local_copy_file("local_file.txt", "/backup/local_file.txt")

                # Видалення локального файлу
                delete_local_file("local_file.txt")

                # Отримання інформації про файл на SMB сервері
                get_file_info(tree, "remote_file.txt")

                # Перевірка доступу до файлу
                check_access_via_smb(tree, "remote_file.txt")
```

```
# Запис даних у файл на SMB сервері
    write_data_to_smb(tree, "remote_file.txt", "New data")

# Читання даних з файлу на SMB сервері
    read_data_from_smb(tree, "remote_file.txt")

# Видалення файлу з SMB сервера
    delete_file_via_smb(tree, "remote_file.txt")

# Викликаємо основну функцію
if __name__ == "__main__":
    main()
```

КБПЗ_2024

```
# Файл: smb_auth.py
# Функція для аутентифікації користувача

import logging

def authenticate_user(username, password):
    logging.info(f"Authenticating user {username}")
    if username == "admin" and password == "password123":
        logging.info(f"User {username} authenticated successfully")
        return True
    else:
        logging.info(f"User {username} authentication failed")
        return False
```

КБПЗ_2024

Файл smb_connection.py

```
# Файл: smb_connection.py
# Функції для підключення до SMB сервера та створення дерева з'єднання

import os
import logging
from smbprotocol.connection import Connection
from smbprotocol.tree import TreeConnect

def connect_to_smb_server(server_ip, username, password):
    try:
        conn = Connection(uuid=os.urandom(16), server_name=server_ip,
username=username, password=password)
        conn.connect()
        logging.info(f"Connected to SMB server at {server_ip}")
        return conn
    except Exception as e:
        logging.error(f"Failed to connect to SMB server: {str(e)}")
        return None

def smb_tree_connect(conn, share_name):
    try:
        tree = TreeConnect(conn, share_name)
        tree.connect()
        logging.info(f"Connected to share {share_name}")
        return tree
    except Exception as e:
        logging.error(f"Failed to connect to share {share_name}: {str(e)}")
        return None
```

Файл smb_operations.py

```
# Файл: smb_operations.py
# Функції для роботи з файлами на SMB сервері

import logging
from smbprotocol.open import Open
from smbprotocol.file_info import FileAttributes, FileBasicInformation,
FileStandardInformation

def transfer_file_via_smb(tree, local_file, remote_file):
    try:
        with open(local_file, 'rb') as f:
            data = f.read()

            smb_file = Open(tree, remote_file)
            smb_file.create(FileAttributes.FILE_ATTRIBUTE_NORMAL)
            smb_file.write(data, 0)
            smb_file.close()

            logging.info(f"File {local_file} transferred to {remote_file} via SMB")
    except Exception as e:
        logging.error(f"Failed to transfer file via SMB: {str(e)}")

def get_file_info(tree, remote_file):
    try:
        smb_file = Open(tree, remote_file)
        smb_file.create(FileAttributes.FILE_ATTRIBUTE_NORMAL)

        file_info = smb_file.query_info(FileBasicInformation)
        logging.info(f"File {remote_file} basic info: {file_info}")

        standard_info = smb_file.query_info(FileStandardInformation)
        logging.info(f"File {remote_file} standard info: {standard_info}")

        smb_file.close()
    except Exception as e:
        logging.error(f"Failed to get file info for {remote_file}: {str(e)}")

def delete_file_via_smb(tree, remote_file):
    try:
        smb_file = Open(tree, remote_file)
        smb_file.create(FileAttributes.FILE_ATTRIBUTE_NORMAL)
        smb_file.delete()
        smb_file.close()

        logging.info(f"File {remote_file} deleted from SMB server")
    except Exception as e:
        logging.error(f"Failed to delete file via SMB: {str(e)}")

def write_data_to_smb(tree, remote_file, data):
    try:
```

```
smb_file = Open(tree, remote_file)
smb_file.create(FileAttributes.FILE_ATTRIBUTE_NORMAL)
smb_file.write(data.encode(), 0)
smb_file.close()

logging.info(f"Data written to {remote_file} on SMB server")
except Exception as e:
    logging.error(f"Failed to write data to {remote_file}: {str(e)}")

def read_data_from_smb(tree, remote_file):
    try:
        smb_file = Open(tree, remote_file)
        smb_file.create(FileAttributes.FILE_ATTRIBUTE_NORMAL)

        data = smb_file.read(1024, 0)
        smb_file.close()

        logging.info(f"Data read from {remote_file} on SMB server")
        return data.decode()
    except Exception as e:
        logging.error(f"Failed to read data from {remote_file}: {str(e)}")
        return None
```

K6П3_2024

Файл local_file_operations.py

```
# Файл: local_file_operations.py
# Функції для роботи з локальними файлами

import os
import shutil
import logging

def create_directory(directory_path):
    try:
        if not os.path.exists(directory_path):
            os.makedirs(directory_path)
            logging.info(f"Directory {directory_path} created")
        else:
            logging.info(f"Directory {directory_path} already exists")
    except Exception as e:
        logging.error(f"Failed to create directory: {str(e)}")

def local_copy_file(source_file, destination_file):
    try:
        shutil.copy(source_file, destination_file)
        logging.info(f"File copied from {source_file} to {destination_file}")
    except Exception as e:
        logging.error(f"Failed to copy file: {str(e)}")

def delete_local_file(file_path):
    try:
        if os.path.exists(file_path):
            os.remove(file_path)
            logging.info(f"File {file_path} deleted")
        else:
            logging.info(f"File {file_path} does not exist")
    except Exception as e:
        logging.error(f"Failed to delete file: {str(e)}")

def check_file_size(file_path):
    try:
        file_size = os.path.getsize(file_path)
        logging.info(f"File size of {file_path}: {file_size} bytes")
        return file_size
    except Exception as e:
        logging.error(f"Failed to check file size: {str(e)}")
        return None

def file_exists(file_path):
    exists = os.path.exists(file_path)
    if exists:
        logging.info(f"File {file_path} exists")
    else:
        logging.info(f"File {file_path} does not exist")
    return exists
```

Файл main.py

```
# Файл: main.py
# Основний файл, що об'єднує всі модулі

from smb_auth import authenticate_user
from smb_connection import connect_to_smb_server, smb_tree_connect
from smb_operations import transfer_file_via_smb, get_file_info,
delete_file_via_smb, write_data_to_smb, read_data_from_smb
from local_file_operations import create_directory, local_copy_file,
delete_local_file, check_file_size, file_exists
from smb_encryption import generate_key, encrypt_file, decrypt_file
from audit import initialize_audit_db, log_user_action
from user_management import initialize_user_db, add_user, authenticate_user_db
from backup import create_backup
from access_control import change_file_permissions, check_file_permissions

def main():
    initialize_user_db()
    initialize_audit_db()

    add_user("admin", "password123")

    if authenticate_user_db("admin", "password123"):
        smb_conn = connect_to_smb_server("192.168.1.1", "admin", "password123")

        if smb_conn:
            tree = smb_tree_connect(smb_conn, "share")

            if tree:
                # Шифруємо файл перед передачею
                generate_key()
                encrypt_file("local_file.txt")

                # Передача зашифрованого файлу через SMB
                transfer_file_via_smb(tree, "local_file.txt", "remote_file.txt")

                # Розшифруємо файл після передачі
                decrypt_file("local_file.txt")

                # Записуємо дії користувача в аудит
                log_user_action("admin", "Transferred file to SMB server")

                # Створення резервної копії
                create_backup("local_file.txt", "/backup")

                # Зміна прав доступу до файлу
                change_file_permissions("local_file.txt", 0o755)

if __name__ == "__main__":
    main()
```

Файл smb_encryption.py

```
# Файл: smb_encryption.py
# Модуль для шифрування та розшифрування файлів

from cryptography.fernet import Fernet
import logging

# Функція для генерації ключа шифрування
def generate_key():
    key = Fernet.generate_key()
    with open("secret.key", "wb") as key_file:
        key_file.write(key)
    logging.info("Encryption key generated and saved as secret.key")

# Функція для завантаження ключа шифрування
def load_key():
    return open("secret.key", "rb").read()

# Функція для шифрування файлу
def encrypt_file(file_path):
    key = load_key()
    fernet = Fernet(key)

    with open(file_path, "rb") as file:
        original = file.read()

    encrypted = fernet.encrypt(original)

    with open(file_path, "wb") as encrypted_file:
        encrypted_file.write(encrypted)

    logging.info(f"File {file_path} encrypted")

# Функція для розшифрування файлу
def decrypt_file(file_path):
    key = load_key()
    fernet = Fernet(key)

    with open(file_path, "rb") as encrypted_file:
        encrypted = encrypted_file.read()

    decrypted = fernet.decrypt(encrypted)

    with open(file_path, "wb") as decrypted_file:
        decrypted_file.write(decrypted)

    logging.info(f"File {file_path} decrypted")
```

Файл audit.py

```
# Файл: audit.py
# Модуль для запису дій користувача в базу даних

import sqlite3
import logging
from datetime import datetime

# Функція для ініціалізації бази даних аудиту
def initialize_audit_db():
    conn = sqlite3.connect('audit_log.db')
    cursor = conn.cursor()

    cursor.execute('''
        CREATE TABLE IF NOT EXISTS audit_log (
            id INTEGER PRIMARY KEY AUTOINCREMENT,
            username TEXT,
            action TEXT,
            timestamp TEXT
        )
    ''')

    conn.commit()
    conn.close()
    logging.info("Audit database initialized")

# Функція для запису дії в журнал
def log_user_action(username, action):
    conn = sqlite3.connect('audit_log.db')
    cursor = conn.cursor()

    timestamp = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    cursor.execute("INSERT INTO audit_log (username, action, timestamp) VALUES
    (?, ?, ?)",
                  (username, action, timestamp))

    conn.commit()
    conn.close()
    logging.info(f"Action logged: {username} performed {action}")
```

Файл user_management.py

```
# Файл: user_management.py
# Модуль для керування користувачами та сесіями

import sqlite3
import logging

# Функція для ініціалізації бази даних користувачів
def initialize_user_db():
    conn = sqlite3.connect('user_management.db')
    cursor = conn.cursor()

    cursor.execute('''
        CREATE TABLE IF NOT EXISTS users (
            id INTEGER PRIMARY KEY AUTOINCREMENT,
            username TEXT UNIQUE,
            password TEXT
        )
    ''')

    conn.commit()
    conn.close()
    logging.info("User management database initialized")

# Функція для додавання користувача
def add_user(username, password):
    try:
        conn = sqlite3.connect('user_management.db')
        cursor = conn.cursor()

        cursor.execute("INSERT INTO users (username, password) VALUES (?, ?)",
            (username, password))

        conn.commit()
        conn.close()
        logging.info(f"User {username} added to database")
    except sqlite3.IntegrityError:
        logging.error(f"User {username} already exists")

# Функція для перевірки аутентифікації
def authenticate_user_db(username, password):
    conn = sqlite3.connect('user_management.db')
    cursor = conn.cursor()

    cursor.execute("SELECT * FROM users WHERE username = ? AND password = ?",
        (username, password))
    user = cursor.fetchone()

    conn.close()

    if user:
        logging.info(f"User {username} authenticated successfully")
```

```
        return True
    else:
        logging.info(f"User {username} authentication failed")
        return False
```

К6П3_2024

Файл backup.py

```
# Файл: backup.py
# Модуль для автоматичного резервного копіювання файлів

import shutil
import logging
from datetime import datetime
import os

# Функція для створення резервної копії файлу
def create_backup(source_file, backup_dir):
    try:
        if not os.path.exists(backup_dir):
            os.makedirs(backup_dir)

        timestamp = datetime.now().strftime("%Y%m%d_%H%M%S")
        backup_file = os.path.join(backup_dir,
f"{timestamp}_{os.path.basename(source_file)}")
        shutil.copy(source_file, backup_file)

        logging.info(f"Backup created: {backup_file}")
    except Exception as e:
        logging.error(f"Failed to create backup: {str(e)}")
```

Файл access_control.py

```
# Файл: access_control.py
# Модуль для управління правами доступу до файлів

import os
import logging

# Функція для зміни прав доступу до файлу
def change_file_permissions(file_path, mode):
    try:
        os.chmod(file_path, mode)
        logging.info(f"Permissions for {file_path} changed to {oct(mode)}")
    except Exception as e:
        logging.error(f"Failed to change permissions for {file_path}: {str(e)}")

# Функція для перевірки прав доступу до файлу
def check_file_permissions(file_path):
    try:
        mode = oct(os.stat(file_path).st_mode)[-3:]
        logging.info(f"Permissions for {file_path}: {mode}")
        return mode
    except Exception as e:
        logging.error(f"Failed to check permissions for {file_path}: {str(e)}")
        return None
```