

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  
“ \_\_\_\_ ” \_\_\_\_\_ 2025 р.

**ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА**  
**за другим (магістерським) рівнем вищої освіти**  
на тему  
**“Дослідження та програмна реалізація системи підвищення**  
**надійності збереження інформації на SSD дисках з**  
**використанням технології RAID”**

Виконав здобувач вищої освіти  
II курсу, групи КІ-24М  
ОПП «Комп’ютерна інженерія»  
спеціальності 123 «Комп’ютерна інженерія»  
\_\_\_\_\_ Хуторний Д.О.  
« \_\_\_\_ » \_\_\_\_\_ 2025 р.

Керівник проекту  
доктор технічних наук, професор  
\_\_\_\_\_ Смірнов О.А.  
« \_\_\_\_ » \_\_\_\_\_ 2025 р.

Рецензент \_\_\_\_\_  
\_\_\_\_\_

## АНОТАЦІЯ

**Хуторний Д.О. Дослідження та програмна реалізація системи підвищення надійності збереження інформації на SSD дисках з використанням технології RAID. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2025.**

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи підвищення надійності збереження інформації на SSD дисках з використанням технології RAID.

Метою розробки є дослідження та програмна реалізація системи підвищення надійності збереження інформації на SSD дисках з використанням технології RAID.

Об'єктом дослідження є процес підвищення надійності збереження інформації на SSD дисках з використанням технології RAID.

Предметом дослідження є методи підвищення надійності збереження інформації на SSD дисках з використанням технології RAID.

Методи дослідження базуються на методах теорії зберігання даних, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи підвищення надійності збереження інформації на SSD дисках з використанням технології RAID.

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 Visual C#.

**Ключові слова:** комп'ютерна інженерія, SSD, RAID

## ABSTRACT

**Khutoryni D.O. Research and software implementation of a system for increasing the reliability of information storage on SSD disks using RAID technology. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2025.**

In this final qualification work for the second (master's) level of higher education, software has been developed, which is intended for a system for increasing the reliability of information storage on SSD disks using RAID technology.

The purpose of the development is the research and software implementation of a system for increasing the reliability of information storage on SSD disks using RAID technology.

The object of the research is the process of increasing the reliability of information storage on SSD disks using RAID technology.

The subject of the research is methods for increasing the reliability of information storage on SSD disks using RAID technology.

The research methods are based on methods of data storage theory, methods of mathematical statistics, and methods of software development.

The result of the work is software implementation of a system for increasing the reliability of information storage on SSD disks using RAID technology.

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 the software are provided.

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

The program was developed in the Visual C# environment.

**Keywords:** computer engineering, SSD, RAID

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ .....	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ .....	7
1.1 Призначення системи.....	7
1.2 Область застосування.....	8
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ .....	10
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	10
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	15
2.3 Розгорнута постановка завдання .....	16
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ .....	18
3.1 Опис функціонування системи .....	18
3.2 Розробка структурної схеми.....	24
3.3 Розробка функціональної схеми .....	30
3.4 Розробка діаграми процесів.....	54
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	56
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	56
4.2 Захист розробленого програмного забезпечення.....	68
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ .....	71
6 НАУКОВА НОВИЗНА .....	74

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>			
<b>Вим.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підп.</b>	<b>Дата</b>	Дослідження та програмна реалізація системи підвищення надійності збереження інформації на SSD диска з використанням технології RAID	<b>Літ.</b>	<b>Аркуш</b>	<b>Аркушів</b>
<b>Розроб.</b>	Хуторний Д.О.					<b>М</b>	1	99
<b>Перев.</b>	Смірнов О.А.					ЦНТУ КІ-24М		
<b>Н.контр.</b>	Коваленко А.С.							
<b>Затв.</b>	Смірнов О.А.							

7	МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ІТ-ПРОЄКТУ .....	75
7.1	Визначення цільової аудиторії кінцевого готового продукту .....	75
7.2	Оцінка привабливості шляхом застосування методів експертних оцінок ...	76
7.3	Вибір методу оцінки вартості ПЗ .....	77
7.4	Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості.....	78
7.5	Пропозиція алгоритму просування проєкту розробки ПЗ .....	79
7.6	Оптимізація каналів збуту та шляхів реалізації ПЗ .....	80
7.7	Визначення ключових факторів успіху конкретного проєкту.....	81
8	ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ .....	82
8.1	Вступ.....	82
8.2	Шкідливі і небезпечні фактори при роботі з комп'ютером.....	83
8.3	Аналіз санітарно-гігієнічних умов праці на робочому місці програміста ...	84
8.4	Розробка заходів з умов поліпшення охорони праці .....	86
8.5	Розрахункова частина .....	87
9	ОСНОВНІ ВИСНОВКИ.....	90
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	92

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ОС	–	Операційна система
ПЗ	–	Програмне забезпечення
BIOS	–	Basic Input-Output System
LINQ	–	Language Integrated Query
RAID	–	Redundant Array of Inexpensive Disks

КБПЗ – 2025

					ВКРМ-123.25.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

## ВСТУП

**Актуальність теми.** Сучасні комп'ютери, ноутбуки та мобільні пристрої використовують твердотільні накопичувачі (SSD), які є енергоефективними та швидкими! Оскільки SSD-накопичувачі не мають рухомих частин, вони також довговічніші, ніж технологія жорстких дисків у мобільних сценаріях.

Але SSD-сховища мають обмеження. Дві основні проблеми:

- Дані можна записати лише обмежену кількість разів.
- Дані не зберігаються надійно протягом тривалого часу, коли живлення вимкнено.

По суті, дані записуються на SSD-накопичувач у вигляді статичних зарядів в окремих комітках. Зазвичай ці комітки утримують заряд дуже довго, але сам процес заряджання комітки є руйнівним. Потрібна висока напруга, щоб послабити бар'єр комітки, перш ніж вона зможе зарядитися. І щоразу, коли записується в комітку, бар'єр постійно послаблюється. Зрештою, комітка не зможе надійно зберігати заряд.

Твердотільні накопичувачі вирішують цю проблему кількома способами. Одна з тактик – вирівнювання зносу, що означає, що дані зазвичай не записуються в одну й ту саму комітку. Накопичувач записує в нові комітки якомога частіше. Це вирівнює знос по всіх комітках. Інша стратегія, яку вони використовують, полягає у збереженні банку додаткових (прихованих) доступних комірок. Коли твердотільний накопичувач бачить, що комітка достатньо «погана», її місце займає одна з «резервних» комірок. Все це відбувається у фоновому режимі.

					ВКРМ-123.25.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

**Мета й завдання дослідження.** Метою роботи є дослідження та програмна реалізація системи підвищення надійності збереження інформації на SSD дисках з використанням технології RAID.

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

– Огляд існуючих систем підвищення надійності збереження інформації на SSD дисках з використанням технології RAID.

– Дослідження системи підвищення надійності збереження інформації на SSD дисках з використанням технології RAID.

– Програмна реалізація системи підвищення надійності збереження інформації на SSD дисках з використанням технології RAID.

*Об'єктом дослідження* є процес підвищення надійності збереження інформації на SSD дисках з використанням технології RAID.

*Предметом дослідження* є методи підвищення надійності збереження інформації на SSD дисках з використанням технології RAID.

*Методи дослідження* базуються на методах теорії зберігання даних, методах математичної статистики, методах розробки програмного забезпечення.

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

– Удосконалено метод підвищення надійності збереження інформації на SSD дисках з використанням технології RAID.

– Розроблено вітчизняний продукт підвищення надійності збереження інформації на SSD дисках з використанням технології RAID, який має більш широкі можливості, на відміну від існуючих аналогів.

**Практична цінність отриманих результатів** полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі підвищення надійності збереження інформації на SSD дисках з використанням технології RAID.

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

**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» (2025 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 підвищення надійності збереження інформації на SSD дисках з використанням технології RAID, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

КБПЗ - 2025

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

# 1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

## 1.1 Призначення системи

Коли елементи втрачають здатність утримувати заряд, першим симптомом є уповільнення зчитування. SSD намагатиметься зчитати комірку, яка іноді повертає неправильне значення (згідно з перевіркою ECC), тому йому доводиться зчитувати її знову, ймовірно, за іншої напруги. Зрештою, комірка повертає правильне значення. Але ці повторні спроби зчитування помітно уповільнюють загальну продуктивність накопичувача.

Для комп'ютерів та SSD-накопичувачів, які залишаються вимкненими протягом тривалого часу, ви побачите пораду, яка рекомендує час від часу вмикати пристрій. Але все, що це насправді робить, це дає SSD можливість позначити пошкоджені комірки, за умови, що пристрій спочатку накаже йому читати або записувати в цю пошкоджену комірку. Деякі високоякісні SSD-накопичувачі періодично перезаписують комірки, щоб самостійно оновлювати дані, але споживчі SSD-накопичувачі зазвичай цього не роблять. Щоб було зрозуміло: включення SSD не перезаряджає комірки і насправді не вирішує ці проблеми.

Нові SSD-накопичувачі можуть надійно зберігати дані протягом кількох років без живлення. Але після активного використання SSD протягом місяців або років має сенс почати періодично оновлювати комірки. Це не тільки забезпечує надійніше зберігання з часом, але й може помітно пришвидшити продуктивність SSD.

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

## 1.2 Область застосування

Твердотільні накопичувачі забезпечують високу надійність для різних застосувань, від базового споживчого обладнання до критично важливих систем. Належні механізми захисту даних можуть максимально збільшити термін служби накопичувача, вживаючи превентивних заходів за потреби та передбачаючи причини втрати даних, такі як надмірний доступ, перегрів, нестабільність живлення та зловмисні атаки.

### Надійність SSD-накопичувачів

Твердотільні накопичувачі (SSD) забезпечують високопродуктивне та компактне масове сховище для вбудованих обчислювальних програм, які роблять можливим сучасний цифровий спосіб життя. Вони використовуються в такому обладнанні, як автомобільні комп'ютерні та навігаційні системи, тонкі клієнти, термінали POS, багатофункціональні принтери, телекомунікаційне обладнання, засоби автоматизації виробництва та серверні системи всіх типів і розмірів, від невеликих центрів обробки даних до хмарних систем.

Типовий термін служби SSD значною мірою залежить від використовуваної технології NAND Flash. QLC (чотирирівнева комірка) та TLC (трирівнева комірка) NAND мають термін служби від 1000 до 3000 циклів програмування/стирання на комірку, тоді як очікуваний термін служби MLC (багатошарова комірка) становить від 3000 до 10 000 циклів. SLC (одношарова комірка) NAND має найдовший термін служби, до 100 000 циклів. Ці відомі обмеження допомагають розробникам систем вибрати оптимальну технологію SSD для свого застосування, залежно від необхідної витривалості, ціни та критичності завдання. TLC NAND, з найнижчою витривалістю, зазвичай має нижчу ціну, ніж інші, і зазвичай використовується у споживчих товарах з обмеженою витратою.

Хоча типовий діапазон терміну служби для будь-якої технології можна передбачити, фактичний термін служби будь-якого диска неможливо знати.

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

Несподіваний збій може призвести до втрати даних, простою системи та витрат на ремонт. Наслідки можуть варіюватися від незручних до катастрофічних. Розробникам систем потрібні інструменти, які можуть допомогти уникнути цих проблем, керуючи та захищаючи SSD-накопичувач, щоб максимізувати термін його служби та вживаючи профілактичних заходів до того, як станеться будь-який збій диска.

Несправний SSD-накопичувач може значно погіршити продуктивність системи. Повільна швидкість читання/запису, збільшення часу доступу та часта обробка помилок можуть призвести до повільної реакції та зниження загальної ефективності системи.

Після збою відновлення та ремонт можуть включати заміну несправного SSD-накопичувача, відновлення даних з резервних копій та переналаштування системи, що може зайняти багато часу та призвести до додаткових витрат. Для діагностики та вирішення проблеми може знадобитися професійна допомога. У деяких випадках, якщо несправність SSD-накопичувача неможливо легко усунути, може знадобитися заміна всієї вбудованої системи.

Щоб пом'якшити наслідки виходу з ладу SSD-накопичувача, важливо мати належні стратегії резервного копіювання, заходи щодо резервування та системи моніторингу.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи підвищення надійності збереження інформації на SSD дисках з використанням технології RAID, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

## 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

#### Серія інтелектуального захисту даних Silicon Motion

Серед інструментів, доступних для безпечного та надійного керування SSD-накопичувачами, вбудовані механізми захисту даних можуть як запобігати, так і пом'якшувати очікувані збої протягом терміну служби. Крім того, вони можуть забезпечити захист від таких небезпек, як збільшення тривалості роботи, надмірний доступ для читання, перехресне нагрівання, перегрівання, нестабільність живлення, злом, і все це може скоротити термін служби накопичувача.

Накопичувачі FerriSSD оснащені набором інтелектуальних механізмів захисту даних, які відповідають потребам зберігання даних різних прикладних систем і можуть бути ефективно активовані в різних середовищах, де може статися втрата даних.

Ці механізми включають IntelligentLog™, який всебічно фіксує стан SSD, щоб розробники систем могли зрозуміти його поточний стан і вжити відповідних заходів. Коли SSD наближається до своєї межі використання, можна проактивно повідомити хост про будь-яку потенційну небезпеку, з точки зору циклів читання та запису, ініціюється міграція даних для забезпечення цілісності даних та запобігання будь-яким втратам. Він також записує основні журнали подій для швидшого та ефективнішого налагодження. Крім того, IntelligentScan™ виконує автоматичне сканування та відновлення, гарантуючи, що дані, що зберігаються в захищеній області, можна правильно зчитувати та записувати. Це збільшує тривалість зберігання даних і продовжує термін служби SSD.

					ВКРМ-123.25.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

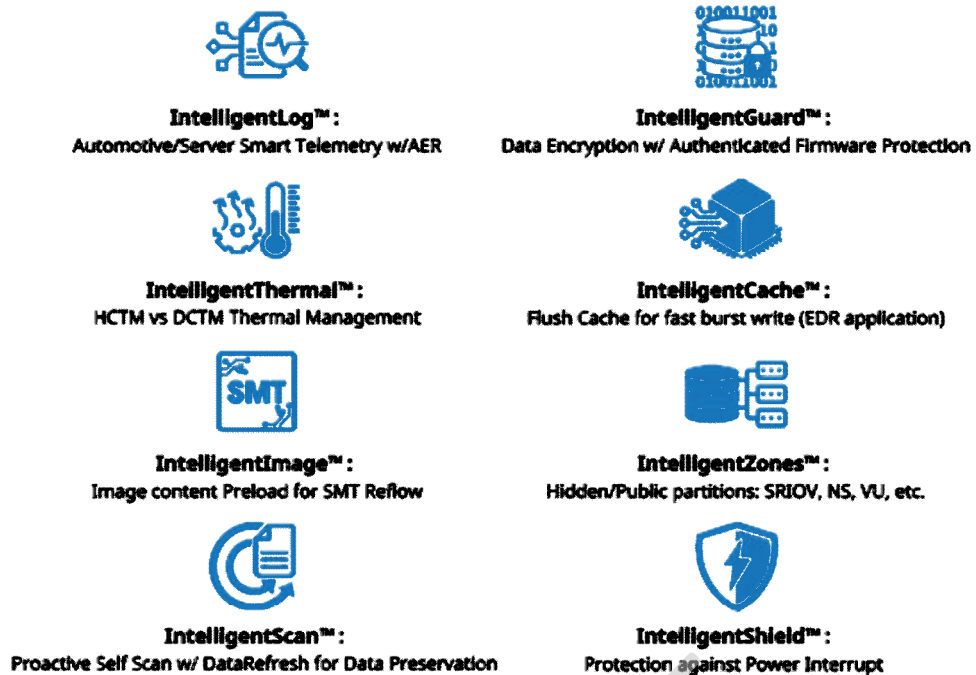


Рисунок 2.1 – Інтерфейс користувача Silicon Motion

### Захист від перегріву

Вплив високих температур може становити низку небезпек для SSD-накопичувачів, зокрема прискорення деградації комірок флеш-пам'яті NAND, що з часом може скоротити термін служби накопичувача. Надмірне нагрівання також може вплинути на цілісність збережених даних і збільшити ризик помилок читання або запису. Також може спостерігатися зниження продуктивності, включаючи уповільнення часу доступу, зниження швидкості передачі даних і проблеми із затримкою, що призводить до зниження загальної продуктивності та швидкості реагування системи.

Щоб запобігти таким проблемам, FerriSSD IntelligentThermal™ контролює температуру накопичувача та активує механізми охолодження, якщо перевищено певний поріг. Він ефективно вибирає між керуванням температурою, керованим хостом (HCTM), та керуванням температурою, керованим накопичувачем (DCTM), щоб контролювати температуру та запобігати пошкодженню SSD в умовах високих температур.

					<b>БКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

## Втрата потужності

Втрата живлення може вплинути на захист даних кількома способами. Якщо живлення раптово втрачається під час операції запису, SSD-накопичувач може не мати достатньо часу для завершення процесу запису даних. Це може призвести до частково записаних даних або невідповідностей метаданих, що призведе до пошкодження або недоступності збереженої інформації. Крім того, втрата живлення під час процесів перерозподілу, таких як вирівнювання зносу, яке виконується регулярно для рівномірного розподілу операцій запису по комірках пам'яті, може призвести до помилок відображення даних або неправильного відстеження розташування даних. Це може спричинити втрату або фрагментацію даних, що вплине на загальну надійність та доступність збережених даних.

Крім того, дані, тимчасово збережені в кеші, можуть бути втрачені, якщо відключення живлення станеться до того, як кешовані дані будуть записані в основне сховище, що призведе до невідповідностей даних або пошкодження файлової системи. Також втрата живлення може перервати оновлення або синхронізацію метаданих, що використовуються для організації та керування збереженими даними, що призведе до пошкодження або невідповідностей, які можуть спричинити проблеми під час завантаження або призвести до втрати цілісності файлової системи.

Для захисту від перебоїв живлення, FerriSSD IntelligentShield™ гарантує, що всі дані, що передаються, безпечно зберігаються у визначеному безпечному місці, запобігаючи втраті даних, спричиненій відключенням живлення. Накопичувач запускає цей механізм, щойно виявляється нестабільність живлення, а також активує IntelligentFlush™ від FerriSSD. Це забезпечує миттєву міграцію даних зі стабільною швидкістю, дозволяючи швидко переміщувати їх до безпечного блоку зберігання (див. рисунок 2.2).

					<b>БКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12



оплавленням, та значно підвищує ефективність виробничої лінії. Нарешті, IntelligentZones™ підтримує конструкцію з кількома просторами імен, що дозволяє налаштовувати SSD відповідно до вимог різних розробників систем. SSD розділений на різні блоки зберігання, а дані розміщуються у відповідному блоці залежно від конструкції. Це забезпечує цілісність даних, а також ефективне використання ємності SSD, тим самим максимізуючи економічні вигоди.

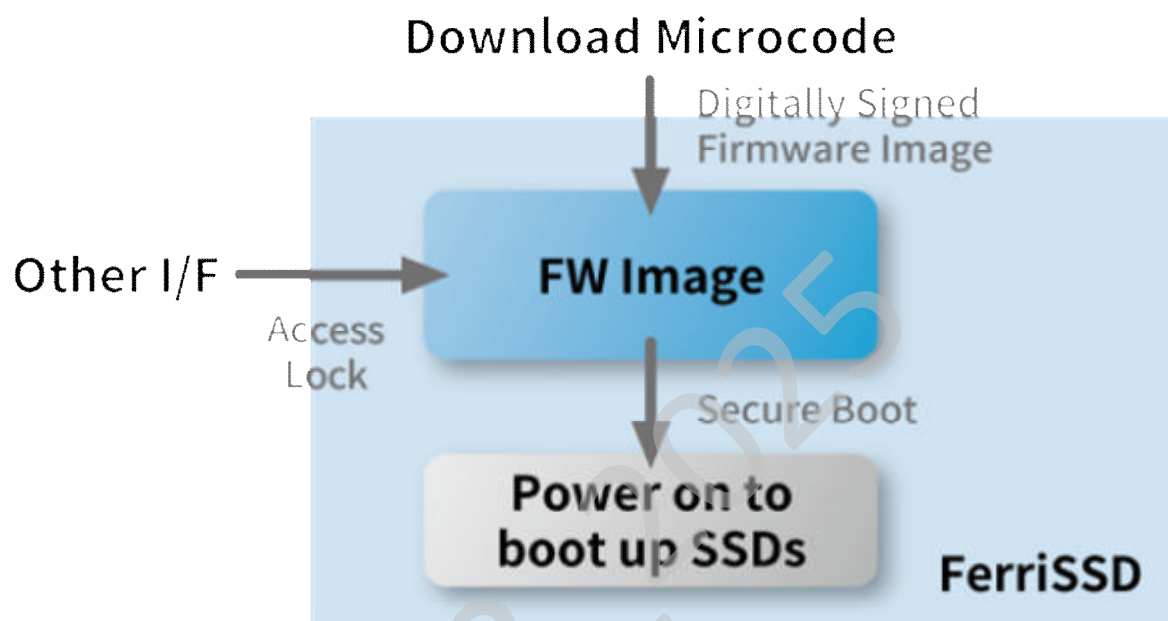


Рисунок 2.3 – Технологія IntelligentGuard™ забезпечує безпеку завантаження, покращуючи механізми автентифікації та захисту мікропрограми SSD

### Висновок

Серія інтелектуального захисту даних забезпечує ефективне рішення для втрати даних у різних середовищах та гарантує безпеку всіх даних, що зберігаються на диску. Завдяки цим передовим заходам захисту користувачі можуть продовжити термін служби своїх накопичувачів і забезпечити постійну безпеку своїх даних.



успадковувати безпосередньо з одного родового класу, але може реалізовувати будь-яке число інтерфейсів. Для методів, які перевизначають віртуальні методи в батьківському класі, необхідно ключове слово `override`, щоб виключити випадкове повторне визначення. У мові Visual C# структура схожа на полегшений клас: це тип, що розподіляється по стопках, що реалізує інтерфейси, але не підтримує спадкування.

### 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 підвищення надійності збереження інформації на SSD дисках з використанням технології RAID.

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 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;

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

- 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 - 2025

					ВКРМ-123.25.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

## 3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

### 3.1 Опис функціонування системи

Індустрія SSD-накопичувачів переживає свій найважливіший період. Те, що починалося як дорога розкіш для техноентузіастів, стало основою всього, від ігрових пристроїв до найбільших у світі центрів обробки даних. Оскільки ринок досяг 61,30 мільярда доларів у 2025 році та стрімко зросте до 129,62 мільярда доларів до 2030 року, ми спостерігаємо революцію в сфері зберігання даних, яка змінює наше уявлення про дані.

Почнемо з того, що всі хочуть знати: наскільки великим стає цей ринок? Численні дослідницькі фірми малюють однакову картину – вибухове зростання в кожному секторі. Mordor Intelligence прогнозує темпи зростання на рівні 16,16% на рік, тоді як інші аналітики ще більш оптимістичні, деякі прогнози сягають майже 280 мільярдів доларів до 2030 року.

Але ось що відрізняє цей період від попередніх технологічних бумів: не лише споживачі є рушійною силою зростання. Корпоративні клієнти лідирують, підживлюючись робочими навантаженнями штучного інтелекту, які просто не можуть функціонувати на традиційних жорстких дисках. Коли ви навчаєте наступний ChatGPT або обробляєте дані автономного транспортного засобу в режимі реального часу, кожна мілісекунда має значення.

Зміна настільки разюча, що навіть найбільш економічно свідомі компанії роблять цей стрибок. Чому? Тому що SSD-накопичувачі виходять з ладу лише з 0,5% порівняно з 5% для традиційних жорстких дисків. Коли прості коштують тисячі за хвилину, ця різниця в надійності стає очевидним бізнес-рішенням.

#### PCIe 5.0

Після років ажіотажу, 2025 рік – це час, коли PCIe 5.0 дійсно став вартим покупки. Ранні користувачі платили високі ціни за накопичувачі, які більшість

					ВКРМ-123.25.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

систем не могли повноцінно використовувати, але масове впровадження знизило витрати, а програмне забезпечення наздогнало.

Ці накопичувачі можуть досягати швидкості 14 000 МБ/с – вдвічі більше, ніж PCIe 4.0 за хороших умов. Для геймерів це означає завантаження величезних відкритих світів за лічені секунди, а не за хвилини. Для професіоналів, які редагують відео 8К, це різниця між плавним відтворенням у реальному часі та постійними заїканнями.

Корпоративні клієнти також звертають на це увагу. Кіохія постачає диски серії SM9 ємністю до 61,44 ТБ – цифри, які здавалися б неможливими ще кілька років тому. Тим часом, PCIe 6.0 залишається на горизонті до 2030 року, що вселяє компаніям впевненість у тому, що їхні інвестиції в PCIe 5.0 не застаріють за одну ніч.

### **Місткість стає смішною**

Пам'ятаєте, коли диск на 1 ТБ здавався величезним? Корпоративні SSD-накопичувачі зараз зазвичай перевищують 100 ТБ, а деякі виробники досягають 512 ТБ за допомогою передової технології виправлення помилок. Це не просто маркетингові трюки – вони вирішують реальні проблеми для компаній, які тонуть у даних.

Секретний інгредієнт – технологія QLC (Quad-Level Cell), яка упаковує чотири біти в кожен комірку пам'яті. Ранні QLC-накопичувачі мали проблеми з довговічністю, але сучасні реалізації значною мірою вирішили ці проблеми, забезпечуючи потужність, яка вимагала б серверних кімнат, заповнених традиційними накопичувачами.

### **Продуктивність, яка дійсно має значення**

Показники швидкості вражають, але реальні покращення продуктивності говорять краще. Сучасні SSD-накопичувачі забезпечують у 30 разів вищу продуктивність, ніж жорсткі диски, у завданнях, які дійсно важливі – час завантаження, запуск програм і передача файлів.

Збільшення енергоспоживання не менш різке. SSD-накопичувачі

					<b>БКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

використовують 2-5 Вт проти 6-15 Вт для жорстких дисків. Це може здатися небагато для одного диска, але помножите це на тисячі серверів, і ви отримаєте серйозні гроші. Центри обробки даних спостерігають падіння рахунків за електроенергію, тоді як продуктивність стрімко зростає.

### **Хто що купує і чому**

#### **Підприємство лідирує**

Корпоративні клієнти становлять 65% ринку SSD, і їхня частка щорічно зростає на 18,2%. Це не імпульсивні покупки – корпоративні покупці мають інші пріоритети, ніж споживачі. Їх цікавлять показники витривалості, гарантійні умови та стабільна продуктивність за високих робочих навантажень.

Значну частину цього попиту забезпечують програми штучного інтелекту. Навчання великих мовних моделей вимагає стабільної пропускної здатності даних, яку традиційні сховища просто не можуть забезпечити. Програмам периферійних обчислень потрібен мілісекундний час відгуку для автономних транспортних засобів та інфраструктури розумного міста.

#### **Споживачі стають витонченими**

Споживчий ринок може бути меншим, займаючи лише 35%, але він стає дедалі складнішим. Геймери розуміють різницю між PCIe 3.0 та 4.0. Творці контенту знають, що їм потрібне швидке сховище для безперебійного редагування 4K.

Для тих, хто хоче оновити старі системи з обмеженим бюджетом, традиційні накопичувачі SATA все ще мають сенс. Такі рішення, як 2,5-дюймовий SATA III SSD серії S330, пропонують значні покращення порівняно з жорсткими дисками, працюючи з існуючим обладнанням. Іноді найкращим оновленням є просто повна відмова від обертання дисків.

#### **Регіональні відмінності мають значення**

Азіатсько-Тихоокеанський регіон домінує з 45% часткою ринку, слугуючи як виробничим центром, так і зростаючою споживчою базою. Південна Корея, Китай та Японія контролюють виробництво, тоді як ринки, що розвиваються,

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

стимулюють зростання впровадження.

Північна Америка становить 30% попиту, на чолі з постачальниками гіпермасштабних хмарних технологій та дослідницькими установами зі штучного інтелекту. Ці клієнти розширюють межі технологій та стимулюють інновації, які зрештою приносять користь усім.

Європа зосереджується на ефективності та відповідності вимогам, причому особливі переваги демонструє автомобільна та промислова галузі. Різні регіони мають різні пріоритети, але всі рухаються до більш продуктивного зберігання даних.

### **Реальні застосування, що стимулюють зростання**

#### **Центри обробки даних трансформують усе**

Постачальники хмарних послуг поспішають замінити жорсткі диски, бо математика переконлива. SSD-накопичувачі забезпечують вищу щільність серверів, знижують витрати на охолодження та забезпечують стабільну продуктивність, з якою жорсткі диски просто не можуть зрівнятися.

Розрахунки рентабельності інвестицій прості: більшість підприємств бачать окупність протягом 18-24 місяців завдяки зниженню витрат на електроенергію, вищим коефіцієнтам консолідації та покращенню продуктивності додатків. Якщо врахувати зниження рівня відмов та витрат на обслуговування, бізнес-кейс стає переконливим.

#### **Штучний інтелект змінює правила гри**

Штучний інтелект – це не просто чергове робоче навантаження, він принципово відрізняється. Навчання великих моделей вимагає постійної пропускної здатності даних, що перевищує 100 ГБ/с. Жорсткі диски розряджаються з набагато меншою швидкістю.

Застосування периферійних обчислень роблять вимоги ще суворішими. Автономні транспортні засоби потребують прийняття рішень у режимі реального часу на основі даних датчиків. Розумні фабрики вимагають миттєвого аналізу виробничих показників. Ці застосунки буквально не можуть функціонувати з

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

традиційними затримками зберігання.

### **Ігри та створення контенту для дорослих**

Технологія DirectStorage від Microsoft є прикладом того, як ігри розвиваються навколо можливостей SSD. Замість того, щоб розробляти ігри з урахуванням обмежень жорсткого диска, нові ігри передбачають швидке сховище та використовують його для реалізації ігрових механік, які раніше були неможливі.

Творцям контенту, які працюють з відео 4K та 8K, потрібен стабільний високошвидкісний доступ до величезних файлів. Сучасні SSD-накопичувачі дозволяють редагувати в режимі реального часу без використання проксі-серверів, що значно підвищує продуктивність творчих професіоналів.

### **Конкуренція на ринку загострюється**

Конкурентне середовище характеризується тим, що відомі гравці борються за частку ринку, тоді як менші компанії знаходять нішеві можливості. Samsung зберігає лідерство завдяки вертикальній інтеграції та передовому виробництву. Western Digital використовує зв'язки з підприємствами та комплексні лінійки продуктів. Micron конкурує завдяки ефективності виробництва та стратегічним партнерствам.

Цікаво, як розвивається конкуренція. Чисто швидкісні війни поступаються місцем інтеграції програмного забезпечення, корпоративним функціям та спеціалізованим рішенням для конкретних застосувань. Компанії створюють комплексні екосистеми, а не просто продають пристрої зберігання даних.

### **Погляд у майбутнє: що далі до 2030 року**

#### **Технологічна дорожня карта**

PCIe 5.0 домінуватиме протягом наступних кількох років, а PCIe 6.0 почне впроваджуватися підприємствами приблизно у 2030 році. Інтерфейси SATA залишаться для бюджетних застосувань та оновлення застарілих систем – не всім потрібна передова продуктивність.

Зростання ємності продовжує неперервний рух у бік 1PB SSD для

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

спеціалізованих гіпермасштабованих застосувань. Але справжній акцент зміщується на стабільну продуктивність за різних робочих навантажень, а не лише на пікову пропускну здатність.

### **Новітні технології**

Технології пам'яті наступного покоління, такі як 3D XPoint, можуть доповнювати флеш-пам'ять NAND для застосувань з наднизькою затримкою. Обчислювальні сховища – SSD-накопичувачі з інтегрованими можливостями обробки – можуть зменшити переміщення даних та підвищити ефективність системи.

Міркування щодо сталого розвитку стають дедалі важливішими. Енергоефективність, можливість переробки та екологічно чисті виробничі процеси впливають на рішення щодо проектування, оскільки зростає екологічна свідомість.

### **Інвестиційна стратегія для бізнесу**

Розумні організації надають пріоритет інвестиціям в інфраструктуру PCIe 5.0 у 2025 році. Це забезпечує запас продуктивності в майбутньому, уникаючи при цьому преміальних витрат та обмеженої підтримки програмного забезпечення, характерних для найсучаснішого PCIe 6.0.

Планування потужностей має передбачати щорічне зростання на 20-30%, щоб врахувати зростаючі потреби в даних, особливо для робочих навантажень штучного інтелекту та аналітики. Але ключовим висновком є оцінка загальної вартості володіння, включаючи споживання енергії, обслуговування та переваги в продуктивності, а не лише витрати на придбання.

### **Підсумок**

Ринок SSD-накопичувачів у 2025 році являє собою зрілу технологію, яка досягає нових висот продуктивності, одночасно розширюючись у застосування, які раніше були непомірно дорогими. Організації, які інвестують у сучасну інфраструктуру зберігання даних сьогодні, позиціонують себе для отримання стійких конкурентних переваг.

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

Ми спостерігаємо конвергенцію робочих навантажень штучного інтелекту, розширення хмарних обчислень та остаточний перехід від механічного зберігання даних. Це створює безпрецедентні можливості для компаній, готових впроваджувати технології зберігання даних наступного покоління.

Фокус змістився з базової заміни жорстких дисків на оптимізацію архітектур сховищ даних для нових робочих навантажень, які вимагають як величезної ємності, так і виняткової продуктивності. Успіх вимагає розуміння не лише поточних технологічних можливостей, але й стратегічних наслідків рішень щодо сховищ даних для довгострокової конкурентоспроможності бізнесу.

Незалежно від того, чи ви оновлюєте ігрову систему, плануєте корпоративну інфраструктуру чи створюєте наступне покоління додатків штучного інтелекту, основа сховища, яку ви оберете сьогодні, визначатиме можливості завтрашнього дня. Революція SSD не настає – вона вже тут, і вона змінює все.

### 3.2 Розробка структурної схеми

Сьогодні навіть мілісекунди мають значення. Моргнуть, і ваш користувач йде до швидшого та витонченішого конкурента. Ми знаходимося на перетині, де продуктивність веб-сайту – це вже не просто жаргон серверної частини; це життєва сила користувацького досвіду, SEO та коефіцієнта конверсії. Повільні сайти дратують. Швидкі перемагають! Крапка. Якщо ви прагнете швидшої роботи сайту та кращої взаємодії користувача, SSD-хостинг – ваша секретна зброя.

На полі цифрової битви, якщо ви не оптимізуєте, ви просто зникаєте. Саме тут і вступає в гру SSD-накопичувач – розкішна необхідність. Це ваша секретна зброя для швидкого веб-хостингу, скорочення часу завантаження та покращення продуктивності веб-сайту. Користувачі навіть не відчувають часу очікування. Погоджуєтесь чи ні, швидкість – це наша валюта. Тепер перейдемо до головного

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

питання.

### **Що таке SSD-сховище у веб-хостингу?**

Пам'ятаєте гудіння та клацання старих комп'ютерів? Це був звук жорстких дисків (HDD) – важких жорстких дисків, що оберталися, як крихітні вінілові платівки, відчайдушно намагаючись отримати ваші дані. Тепер же тиша (спокій)... і швидкість. Це SSD-накопичувачі – твердотільні накопичувачі – сучасне диво без рухомих частин, лише миттєвий доступ до ваших файлів, як перехід з рикші на швидкісний поїзд.

У світі веб-хостингу це не просто оновлення, це революція. SSD-накопичувачі у веб-хостингу означають, що ваш веб-сайт завантажується швидше, обробляє більше користувачів і рідше аварійно завершує роботу. Жорсткі диски були непогані, коли у нас було терпіння (і dial-up). Але в якому ми сьогодні? Користувачі відмовляються від доступу, якщо ваш сайт затримується навіть на 2 секунди. SSD-накопичувачі отримують доступ до даних миттєво, а це означає, що ваші зображення, скрипти та бази даних доставляються блискавично.

Хостинг на SSD – це вже не бонус, а виживання. У світі, залежному від миттєвого задоволення, повільне завантаження – це вірне цифрове самогубство.

### **SSD проти HDD у хостингу: технічне порівняння**

Жорсткі диски (HDD) (накопичувачі вашого дідуся) використовують обертові магнітні диски та механічну ручку для читання/запису даних. Уявіть собі це як програвач платівок. Механічний, ностальгічний... і болісно повільний. З іншого боку, SSD-накопичувачі – це круті дітища – без рухомих частин, лише чиста електронна магія, яка майже миттєво отримує доступ до даних. Як перемикання вимикача світла замість того, щоб чекати, поки закипить чайник.

Цифри? Жорстокі. SSD-накопичувачі можуть бути до 20 разів швидшими, зі значно меншою затримкою, надзвичайно високою кількістю операцій вводу/виводу за секунду (IOPS) та швидшим часом доступу до даних. Крім того, вони міцніші, відсутність рухомих частин означає менше збоїв, і вони енергоємні,

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

на відміну від енергоємних жорстких дисків (знову ж таки, фу!).

У веб-хостингу це означає, що швидкість веб-сайту з SSD не просто помітна, а й змінює правила гри. Швидше завантаження сторінок, швидші виклики бази даних, плавніший користувацький досвід. Тим часом жорсткі диски вагаються, як сонний динозавр.

Досі дивуєтесь, чому SSD кращий за HDD для хостингу? Простіше кажучи: швидший час доступу, менше збоїв та неперевершена стабільність продуктивності в сценаріях з високим трафіком.

Тож так, обидва зберігають дані. Але один проноситься, як Flash, поки інший ще зав'язує шнурки. Вибирайте з розумом, або ризикуєте втратити відвідувачів ще до того, як ваша головна сторінка завантажиться.

### **Переваги SSD-хостинга для власників веб-сайтів**

Перехід на SSD-хостинг – це не просто «приємна річ», це веб-еквівалент відмови від кінного воза на користь ракети. Ось чому власники сайтів усюди обирають цей варіант:

– Швидше завантаження веб-сайту. Якщо ви керуєте інтернет-магазином, переваги SSD-хостингу для веб-сайтів електронної комерції ще більш важливі. Кожна секунда, скорочена під час завантаження, може безпосередньо вплинути на ваші продажі. Швидкість – це головне, а SSD-сховище – головне.

– SSD проти HDD у хостингу. На відміну від старих жорстких дисків (HDD), які бавляться з даними, SSD-накопичувачі доставляють контент миттєво. На практиці? Ваш показник відмов падає, залученість зростає, а клієнти не йдуть, поки не завантажиться ваш логотип. Якщо вам потрібен швидкий веб-хостинг і така швидкість веб-сайту з SSD, що користувачі кажуть «Вау», це ваш шлях. Для електронної комерції, портфоліо або навіть того блогу про котів, який ви думаєте створити, швидкість має значення.

– Підвищена надійність та цілісність даних. Жорсткі диски мають рухомі частини, і знаєте що? Рухомі елементи зрештою виходять з ладу. SSD-накопичувачі, без механічних частин, схожі на цифрові танки: тихі, міцні та

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

набагато рідше ламаються. Це означає кращий час безвідмовної роботи, менше помилок даних та більше спокою для вас. Якщо ваш сайт – це ваш бізнес, ви не можете дозволити собі збоїв. Одного збою забагато, а SSD-накопичувачі допомагають переконатися, що цього не станеться.

– Покращена SEO-продуктивність. У Google є одне правило: пришвидшуйся або відставай. Сайти на SSD завантажуються швидше, що подобається як користувачам, так і володарям пошукових систем. Швидкість завантаження сторінки – це відомий сигнал ранжування, і SSD допомагає вам впевнено поставити цю галочку. Якщо ви серйозно налаштовані піднятися в результатах пошуку, інвестування в SSD – це не просто варіант, це одна з найрозумніших переваг SSD-хостингу, яку ви можете використовувати. Швидке завантаження сторінок на SSD-хостингу також знижує показник відмов, що є ще одним непрямим посиленням SEO, особливо важливим для блогів, що розвиваються, та веб-сайтів малого бізнесу, які прагнуть піднятися в рейтингу:

– Менше споживання ресурсів та ефективність хостингу. SSD-накопичувачі не просто швидкі, вони ефективні. Ви обслуговуєте користувачів з меншим навантаженням на процесор, краще обробляєте трафік та швидше обробляєте дані. Платформи CMS, такі як WordPress, Magento та інші, процвітають у своїх SSD-середовищах. Це означає плавніше кешування, швидші виклики бази даних та більшу віддачу від витрат на хостинг. Результат? Ви покращуєте продуктивність веб-сайту, не витрачаючи ресурси.

### **Реальні переваги продуктивності завдяки хостингу на SSD-дисках**

Давайте на секунду відкинемо теорію та поговоримо про холодні, однозначні результати. Коли веб-сайти переходять на SSD-хостинг, різниця не є ледь помітною – вона вражає. Дослідження показують, що веб-сайти, розміщені на SSD-серверах, мають скорочення часу завантаження до 50%. Це вдвічі менше очікування, вдвічі більше задоволення. Уявіть, що ви скорочуєте час завантаження на 3 секунди з 6-секундного – тепер ви граєте у вищій лізі швидкого веб-хостингу.

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

Візьмемо, наприклад, WordPress. Це потужна CMS, але вона також багато працює з базами даних. На SSD-хостингу швидкість виконання запитів значно вища, що робить адміністративну панель більш адаптивною, а фронтенд – оперативнішим. Те саме стосується таких платформ, як Joomla, Magento або Drupal – SSD спрощує їх усі.

Чи то інтернет-магазин, який завантажує тисячі зображень товарів, чи блог зі стрімким зростанням трафіку, переваги SSD-хостинга очевидні: краща швидкість, кращий досвід та помітна перевага в сучасному інтернеті, де швидкість залежить від сайту.

### **Чому сайти електронної комерції потребують SSD-хостинга**

Не кожному вебсайту потрібна прискорена швидкість, але якщо ви запускаєте щось серйозне, вам це точно потрібно.

– Вебсайти електронної комерції? Вони процвітають за рахунок швидкості, а повільне оформлення замовлення вбиває продажі.

– Сайти з високим навантаженням, завантажені зображеннями, відео або великим трафіком? SSD-накопичувачі з'їдають це навантаження на сніданок!

– SaaS-платформи чи недбалі стартапи? SSD-хостинг забезпечує продуктивність, швидкість та стійкість до збоїв, коли кількість користувачів раптово зростає.

Це ті самі поля битви, де продуктивність має найбільше значення, і SSD-накопичувачі чудово справляються з цим завданням. Якщо ви серйозно ставитеся до масштабування та хочете дійсно покращити продуктивність веб-сайту, переваги SSD-хостинга говорять самі за себе.

### **Як вибрати правильного постачальника SSD-хостингу**

Вибір правильного SSD-сховища для веб-хостингу – це не просто вибір швидшого варіанту. Шукайте постачальника, який пропонує щедрі ліміти сховища, підтримку NVMe SSD (наступне покоління швидкісних SSD), стратегічно розташовані центри обробки даних та надійну гарантію безвідмовної роботи, адже навіть мілісекунда простою може дорого коштувати вашому

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

бізнесу. Ретельно збалансуйте ціну та продуктивність; найдешевший варіант може заощадити копійки, але згодом погіршити продуктивність.

Не менш важливими є підтримка та масштабованість. Переконайтеся, що постачальник пропонує цілодобову технічну підтримку та інфраструктуру для зростання відповідно до потреб вашого бізнесу.

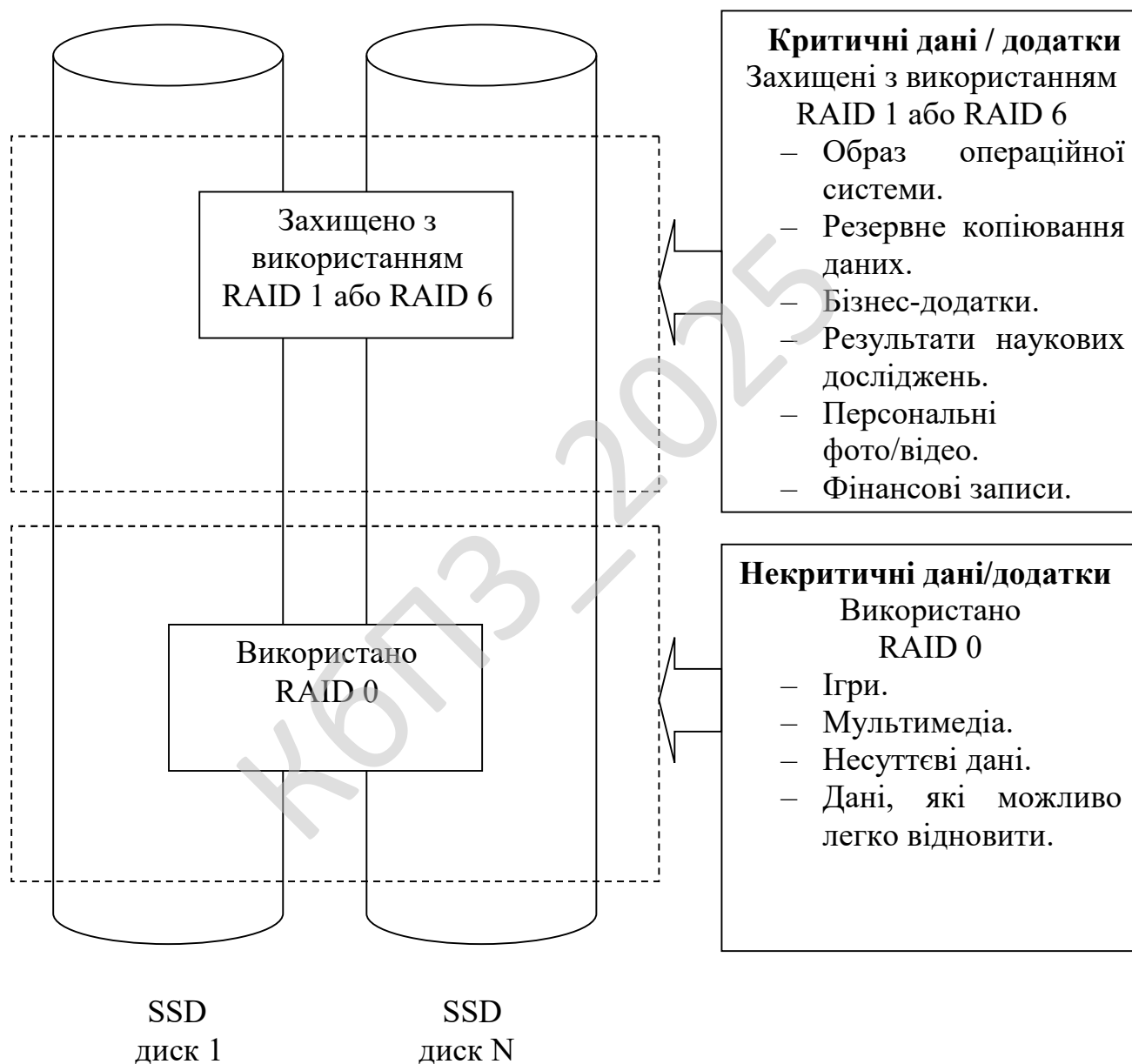


Рисунок 3.1 – Структурна схема розробленої системи

## Майбутні тенденції: SSD-хостинг та розвиток Інтернету

Інтернет не сповільнюється, і ваш хостинг теж не повинен. Майбутнє явно схиляється до твердотільних накопичувачів на базі NVMe та PCIe, які пропонують блискавично високу швидкість передачі даних, що набагато перевищує традиційні твердотільні накопичувачі SATA. Зі зростанням хмарного хостингу та периферійних обчислень потреба у високошвидкісному сховищі з низькою затримкою стає критичною, і твердотільні накопичувачі є лідерами в цьому питанні.

У цій ситуації питання SSD проти HDD у хостингу вже навіть не обговорюється: HDD швидко стають пережитком минулого. Переваги SSD-хостингу – швидкість, масштабованість та ефективність – формують наступне покоління Інтернету. Будьте попереду або забудьте.

Структурна схема розробленої системи зображена на рисунку 3.1.

### 3.3 Розробка функціональної схеми

Розрізняють кілька основних рівнів RAID-масивів: RAID 0, 1, 2, 3, 4, 5, 6, 7. Також існують комбіновані рівні, такі як RAID 10, 0+1, 30, 50, 53 і т.п. Розглянемо принципи функціонування, достоїнства й недоліки основних рівнів.

#### **RAID 0 – Дисконий масив без відказостійкості (Striped Disk Array without Fault Tolerance)**

Дисконий масив без надлишкового зберігання даних. Інформація розбивається на блоки, які одночасно записуються на окремі диски, що забезпечує підвищення продуктивності. Такий спосіб зберігання інформації ненадійний, оскільки поломка одного диска приводить до втрати всієї інформації, тому рівнем RAID як таким не є.

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

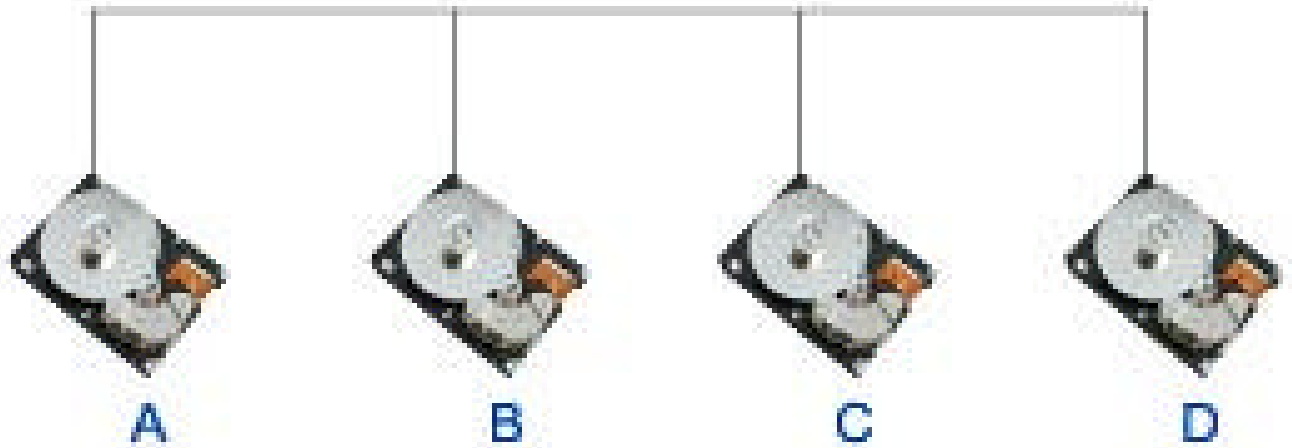


Рисунок 3.2 – Схема RAID 0

RAID 0 – дешевий і продуктивний, але ненадійний. За рахунок можливості одночасного вводу/виводу з декількох дисків масиву RAID 0 забезпечує максимальну швидкість передачі даних і максимальну ефективність використання дискового простору, тому що не потрібно місця для зберігання контрольних сум. Реалізація цього рівня дуже проста. RAID 0, як правило, застосовується в тих областях, де потрібна швидка передача великого обсягу даних. Для реалізації масиву потрібно не менше двох вінчестерів.

Переваги:

- найвища продуктивність у додатках, що вимагають інтенсивної обробки запитів вводу/виводу й даних великого обсягу;
- простота реалізації;
- низька вартість;
- максимальна ефективність використання дискового простору – 100%.

Недоліки:

- не є "сьогоденням" RAID'ом, оскільки не підтримує відказостійкість;
- відмова одного диска спричиняє втрату всіх даних масиву.





– при збільшенні кількості дисків накладні витрати зменшуються.

Недоліки:

- низька швидкість обробки запитів;
- висока вартість;
- більша надмірність.

### **RAID 3 – Відказостійкий дисковий масив з паралельною передачею даних і парністю (Parallel Transfer Disks with Parity)**

Відказостійкий масив з паралельним вводом/виводом даних і диском контролю парності. Потік даних розбивається на порції на рівні байт (хоча можливо й на рівні біт) і записується одночасно на всі диски масиву, крім одного. Один диск призначений для зберігання контрольних сум, що обчислюються при записі даних. Поломка кожного з дисків масиву не приведе до втрати інформації.

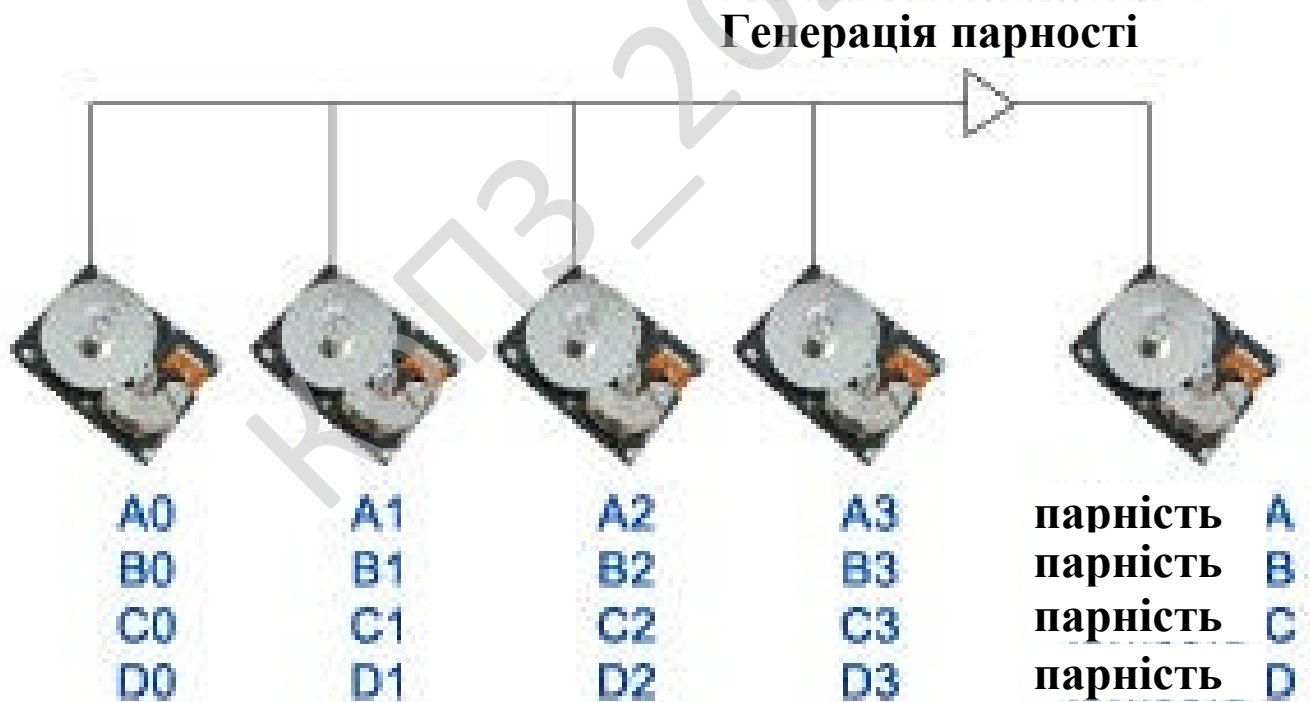


Рисунок 3.5 – Схема RAID 3

В RAID 3 інформація розбивається на порції однакового розміру.

Цей рівень має набагато меншу надмірність, чим RAID 2. У RAID 2 більшість дисків, що зберігають контрольну інформацію, потрібні для визначення несправного розряду. Як правило, RAID-контролери можуть одержати дані про помилку за допомогою механізмів відстеження випадкових збоїв. За рахунок розбивки даних на порції RAID 3 має високу продуктивність. Оскільки при кожній операції вводу/виводу виробляється обіг практично до всіх дисків масиву, те одночасна обробка декількох запитів неможлива.

Цей рівень підходить для додатків з файлами великого обсягу й малою частотою обігів (в основному це сфера мультимедіа). Використання тільки одного диска для зберігання контрольної інформації пояснює той факт, що коефіцієнт використання дискового простору досить високий (як наслідок цього – відносно низька вартість). Для реалізації масиву потрібно не менше трьох вінчестерів.

Переваги:

- відмова диска мало впливає на швидкість роботи масиву;
- висока швидкість передачі даних;
- високий коефіцієнт використання дискового простору.

Недоліки:

- складність реалізації;
- низька продуктивність при великій інтенсивності запитів даних невеликого обсягу.

#### **RAID 4 – Відказостійкий масив незалежних дисків із загальним диском парності (Independent Data Disks with Shared Parity Disk)**

Цей масив дуже схожий на рівень RAID 3. Потік даних розділяється не на рівні байтів, а на рівні блоків інформації, кожний з яких записується на окремий диск. Після запису групи блоків обчислюється контрольна сума, що записується на виділений для цього диск.

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35





В RAID 5 відсутній виділений диск для зберігання інформації про парність.

Найбільший недолік рівнів RAID від 2-го до 4-го – це наявність окремого диска (або дисків), що зберігає інформацію про парність. Швидкість виконання операцій зчитування досить висока, тому що не вимагає звертання до цього диска. Але при кожній операції запису на ньому змінюється інформація, тому схеми RAID 2-4 не дозволяють проводити паралельні операції запису. RAID 5 не має цього недоліку, тому що контрольні суми записуються на всі диски масиву, що уможлиблює виконання декількох операцій читання або записи одночасно. RAID 5 має досить високу швидкість запису/читання й малу надмірність.

Переваги:

- висока швидкість запису даних;
- досить висока швидкість читання даних;
- висока продуктивність при великій інтенсивності запитів читання/запису даних;
- високий коефіцієнт використання дискового простору.

Недоліки:

- низька швидкість читання/запису даних малого обсягу при одиничних запитах;
- досить складна реалізація;
- складне відновлення даних.

### **RAID 6 – Відказостійкий масив незалежних дисків із двома незалежними розподіленими схемами парності (Independent Data Disks with Two Independent Distributed Parity Schemes)**

RAID 6 – це відказостійкий масив незалежних дисків з розподілом контрольних сум, обчислених двома незалежними способами. Цей рівень багато в чому схожий з RAID 5. Тільки в ньому використовується не одна, а дві незалежні схеми контролю парності, що дозволяє зберігати працездатність системи при одночасному виході з ладу двох накопичувачів. Для обчислення контрольних сум

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

в RAID 6 використовується алгоритм, побудований на основі коду Ріда-Соломона (Reed-Solomon).

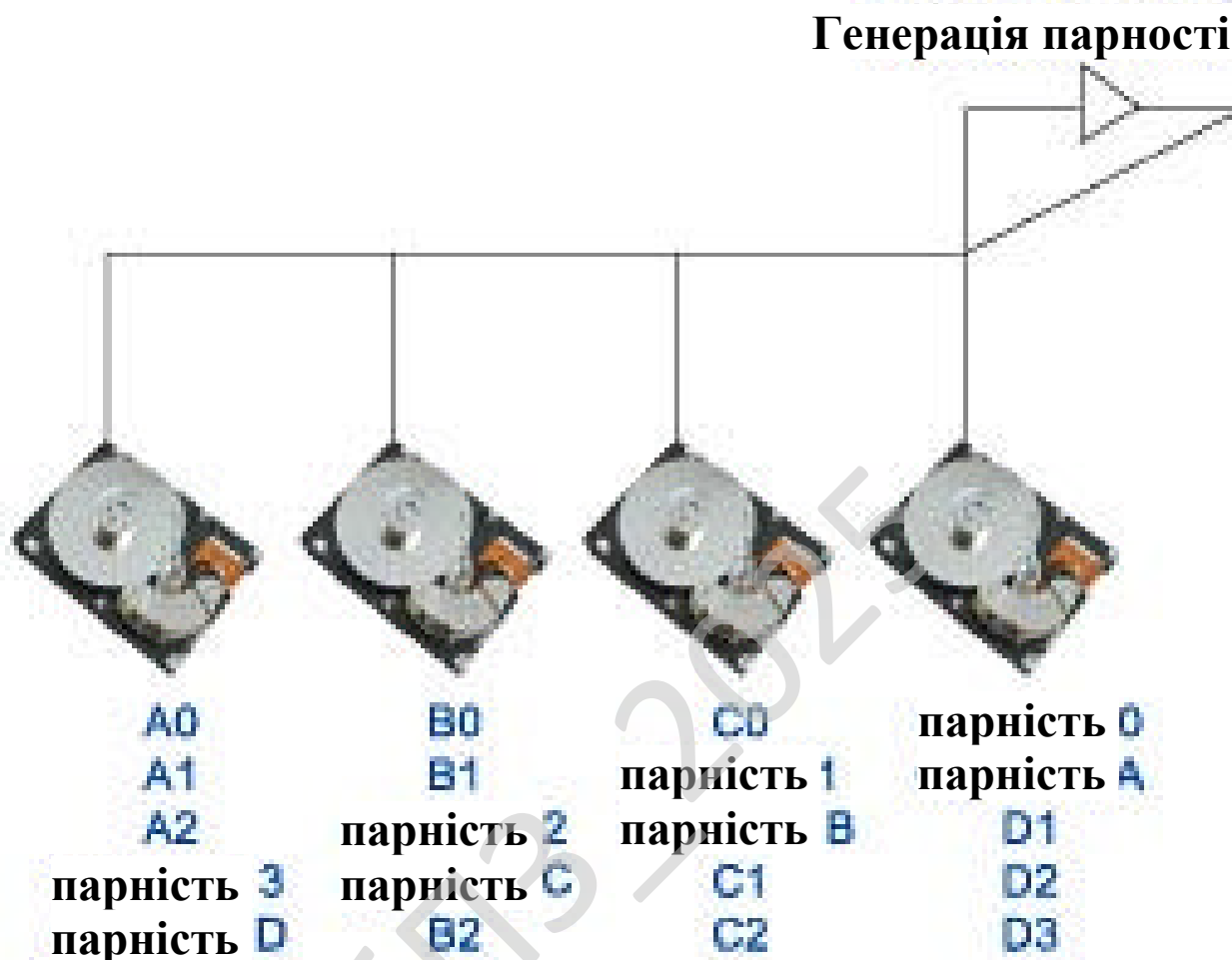


Рисунок 3.8 – Схема RAID 6

RAID 6 використовує дві незалежні схеми контролю парності.

Цей рівень має дуже високу відказостійкість, більшу швидкість зчитування (дані зберігаються блоками, немає виділених дисків для зберігання контрольних сум). У той же час через великий обсяг контрольної інформації RAID 6 має низьку швидкість запису. Він дуже складний у реалізації, характеризується низьким коефіцієнтом використання дискового простору: для масиву з п'яти дисків він становить усього 60%, але з ростом числа дисків ситуація виправляється.

RAID 6 по багатьом характеристикам програє іншим рівням, тому на сьогодні не одержав комерційного застосування.

Переваги:

- висока відказостійкість;
- досить висока швидкість обробки запитів;

Недоліки:

- низька швидкість читання/запису даних малого обсягу при одиничних запитах;
- дуже складна реалізація;
- складне відновлення даних;
- низька швидкість запису даних.

**RAID 7 – Відказостійкий масив, оптимізований для підвищення продуктивності (Optimized Asynchrony for High I/O Rates as well as High Data Transfer Rates)**

На відміну від інших рівнів, RAID 7 не є відкритим індустріальним стандартом – це зареєстрована торговельна марка компанії Storage Computer Corporation. Масив ґрунтується на концепціях, використаних у третьому й четвертому рівнях. Додалася можливість кешування даних. До складу RAID 7 входить контролер з убудованим мікропроцесором під керуванням операційної системи реального часу (real-time OS). Вона дозволяє обробляти всі запити на передачу даних асинхронно й незалежно.

RAID 7 – зареєстрована торговельна марка компанії Storage Computer Corporation.

Блок обчислення контрольних сум інтегрований із блоком буферізації; для зберігання інформації про парність використовується окремий диск, що може бути розміщений на будь-якому каналі. RAID 7 має високу швидкість передачі даних і обробки запитів, гарну масштабованість. Самим більшим недоліком цього рівня є вартість його реалізації.

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40



## RAID 10

RAID 10 – відказостійкий масив з дублюванням і паралельною обробкою. Ця архітектура виявляє собою масив типу RAID 0, сегментами якого є масиви RAID 1. Він поєднує в собі високу відказостійкість і продуктивність.

### Комбіновані рівні

Крім базових рівнів RAID 0 – RAID 5, описаних у стандарті, існують комбіновані рівні RAID 1+0, RAID 3+0, RAID 5+0, RAID 1+5, які різні виробники інтерпретують кожний по-своєму.

– RAID 1+0 – це сполучення дзеркалювання й чергування.

Нинішні контролери використовують цей режим за замовчуванням для RAID 1. Тобто, 1 диск основний, 2-й диск – дзеркало, причому читання виробляється з них по черзі, як для RAID 0. Властиво, зараз можна вважати що RAID 1 і RAID 1+0 – це просто різні назви того самого методу апаратного дзеркалювання дисків. Але не варто забувати, що повноцінний RAID 1+0 повинен містити як мінімум 4 диски.

– RAID 5+0 – це чергування томів 5-го рівня. RAID 1+5 – RAID 5 із дзеркальованою парою.

Комбіновані рівні успадковують як переваги, так і недоліки своїх «батьків»: поява чергування в рівні RAID 5+0 анітрошки не додає йому надійності, але зате позитивно відбивається на продуктивності. Рівень RAID 1+5, напевно, дуже надійний, але не найшвидший і, до того ж, украй неекономічний: корисна ємність тому менше половини сумарної ємності дисків.

Варто відзначити, що кількість жорстких дисків у комбінованих масивах також зміниться. Наприклад для RAID 5+0 використовують 6 або 8 жорстких дисків, для RAID 1+0 – 4, 6 або 8.

### Matrix RAID

Matrix RAID – це технологія, реалізована фірмою Intel у своїх чипсетах починаючи з ICH6R. Строго говорячи, ця технологія не є новим рівнем RAID (її аналог існує в апаратних RAID-контролерах високого рівня), вона дозволяє,

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

використовуючи невелику кількість дисків організувати одночасно один або кілька масивів рівня RAID 1, RAID 0 і RAID5. Це дозволяє за порівняно невеликі гроші забезпечити для одних даних підвищену надійність, а для інших високу швидкість доступу.

### **Програмний RAID**

Для реалізації RAID можна застосовувати не тільки апаратні засоби, але й повністю програмні компоненти (драйвери). Наприклад, у системах на ядрі Linux існують спеціальні модулі ядра, а управляти RAID-пристроями в GNU/Linux можна за допомогою утиліти mdadm. Програмний RAID має свої достоїнства й недоліки. З одного боку, він нічого не коштує (на відміну від апаратних RAID-контролерів, ціна яких від \$250). З іншого боку, програмний RAID використовує ресурси центрального процесора, і в моменти пікового навантаження на дискову систему процесор може значну частину потужності витратити на обслуговування RAID-пристроїв.

Ядро GNU/Linux 2.6.28 (останнє з вийшли в 2008 році) підтримує програмні RAID наступних рівнів: 0, 1, 4, 5, 6, 10. Реалізація дозволяє створювати RAID на окремих розділах дисків, що аналогічно описаному вище Matrix RAID. Завантаження підтримується (grub, lilo) тільки з дисків з RAID 1.

ОС Windows 2007/2008/2010 підтримує програмний RAID 0, RAID 1 і RAID 5. Більш точно, Windows XP Pro підтримує RAID 0. Підтримка RAID 1 і RAID 5 заблокована розроблювачами, але, проте, може бути включена, шляхом редагування системних бінарних файлів ОС. Windows Server 2003 – 0, 1 і 5. Windows XP Home RAID не підтримує.

В ОС FreeBSD є кілька реалізацій програмного RAID. Так, atactrol, може як повністю будувати програмний RAID, так і може підтримувати напівпаратний RAID на таких чипах як ICH5R. В FreeBSD, починаючи з версії 5.0, дискова підсистема управляється убудованим у ядро механізмом GEOM. GEOM надає модульну дискову структуру, завдяки якій народилися такі модулі як gstripe (RAID 0), gmirror (RAID 1), gRAID3 (RAID 3), gconcat (об'єднання

декількох дисків у єдиний дисковий розділ). Так само існують застарілі класи ccd (RAID 0, RAID 1) і gvinum (менеджер логічних томів vinum). Починаючи з FreeBSD 7.0 підтримується файлова система ZFS у якій можна збирати наступні рівні RAID: 0, 1, 5, 6, а так само комбинуємі рівні.

OpenSolaris і Solaris 10 використовують Solaris Volume Manager, що підтримує RAID-0, RAID-1, RAID-5 і будь-які їхні комбінації як 1+0. Підтримка RAID-6 здійснюється у файловій системі ZFS.

Система підвищення надійності збереження інформації на SSD дисках з використанням технології RAID, яка розроблена у результаті виконання дипломного проектування, будується на використанні технології Intel Intel® SATA Solid-State Drive, з використанням RAID 0 та RAID 1 або RAID 6, в залежності від кількості дисків у RAID-масиві. Якщо диска два, то використовується RAID 1, якщож їх більше то використовується RAID 6.

RAID 0 дозволяє розбивати інформацію на блоки, які одночасно записуються на окремі диски, що забезпечує підвищення продуктивності. Такий спосіб зберігання інформації ненадійний, оскільки поломка одного диска приводить до втрати всієї інформації.

При використанні RAID 1 два накопичувачі містять однакову інформацію і є одним логічним диском. При виході з ладу одного диска його функції виконує інший. Для реалізації масиву потрібно не менше двох вінчестерів.

RAID 6 дозволяє сформувати відказостійкий масив незалежних дисків з розподілом контрольних сум, обчислених двома незалежними способами. В ньому використовується дві незалежні схеми контролю парності, що дозволяє зберігати працездатність системи при одночасному виході з ладу двох накопичувачів. Для обчислення контрольних сум в RAID 6 використовується алгоритм, побудований на основі коду Ріда-Соломона (Reed-Solomon).

На рисунку 3.10 зображена функціональна схема системи. Нижче розглянемо її більш докладно.

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

Система складається з наступних блоків:

– Блок розподілу інформації за критерієм критичності/важливості визначає який саме вид RAID-масиву необхідно буде використовувати. Якщо інформація не є критичною то буде використовуватися RAID 0. Якщо інформація є важливою, то в залежності від ступеня важливості, та вимог до швидкодії, буде використовуватися або RAID 1, який є більш швидким, та менш надійним, або RAID 6, який є дуже надійним але менш швидким.

– Блок вибору кількості дисків – призначений для визначення кількості дисків з яких буде складатися RAID-масив. Якщо дисків буде 2 то буде використовуватися технологія RAID 0 та RAID 1. Якщо дисків буде більше 2 то буде використовуватися технологія RAID 0 та RAID 6.

– Блок формування віртуальних дисків використовується у разі коли диск тільки один. Тоді на цьому диску формуються декілька віртуальних дисків, з якими відбуваються дії аналогічні як ті, що відбуваються над фізичними дисками. Тобто RAID-масив формується з віртуальних дисків.

Блоки RAID 0, RAID 1 та RAID 6 реалізують ту, або іншу технологію.

У блоці RAID 6 використовується кодек Ріда-Соломона. Нижче надамо опис цього кодеку.

**Коди Ріда-Соломона** – недвійкові циклічні коди, що дозволяють виправляти помилки в блоках даних. Елементами кодового вектора є не біти, а групи біт (блоки). Дуже поширені коди Ріда-Соломона, що працюють із байтами (октетами).

У цей час широко використовується в системах відновлення даних з компакт-дисків, при створенні архівів з інформацією для відновлення у випадку ушкоджень, у завадостійкому кодуванні.

Коди Ріда-Соломона є важливим частковим випадком БЧХ-коду, корінь полінома, що породжує, якого лежать у тім же полі, над яким і будується код ( $m = 1$ ).

Коди Боуза-Чоудхури-Хоквінгхема (БЧХ коди) – у теорії кодування це

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

широкий клас циклічних кодів, застосовуваних для захисту інформації від помилок. Відрізняється можливістю побудови коду із заздалегідь певними коригувальними властивостями, а саме, мінімальною кодовою відстанню.

### Поліном, що породжує

Визначення поліномом, що породжує, циклічного  $(n, k)$  коду  $C$  називається такий ненульовий  $g(x) = \sum_{i=0}^r g_i x^i$  поліном з  $C$ , ступінь якого найменша й коефіцієнт при старшому ступені  $g_r = 1$ .

### Теорема 3.1

Якщо  $C$  – циклічний  $(n, k)$  код і  $g(x)$  – його поліном, що породжує, тоді ступінь  $g(x)$  дорівнює  $r = n - k$  і кожне кодове слово може бути єдиним чином представлено у вигляді  $c(x) = m(x)g(x)$ , де ступінь  $m(x)$  менше або дорівнює  $k - 1$ .

### Теорема 3.2

$g(x)$  – поліном, що породжує, циклічного  $(n, k)$  коду є дільником двочлена  $x^n - 1$ .

**Наслідок:** у такий спосіб як поліном, що породжує, можна вибирати будь-який поліном, дільник  $x^n - 1$ . Ступінь обраного полінома буде визначати кількість перевірочних символів  $r$ , число інформаційних символів  $k = n - r$ .

### Матриця, що породжує

Поліноми  $g(x), xg(x), x^2g(x), \dots, x^{k-1}g(x)$  лінійно незалежні, інакше  $m(x)g(x) = 0$  при ненульовому  $m(x)$ , що неможливо.

Значить кодові слова можна записувати, як і для лінійних кодів, наступним чином:

$$\bar{m}G = (m_0, m_1, \dots, m_{k-1}) \begin{bmatrix} g(x) \\ xg(x) \\ \dots \\ x^{k-1}g(x) \end{bmatrix} = m(x)g(x), \quad (3.1)$$

де  $G$  є матрицею, що породжує,  $m(x)$  – інформаційним поліномом.

Матрицю  $G$  можна записати в символній формі:



довжиною  $t$  і менше. Відповідно до теореми про границю Рейгера, коди Ріда-Соломона є оптимальними з погляду співвідношення довжини пакета й можливості виправлення помилок – використовуючи  $2t$  додаткових перевірочних символів виправляються  $t$  помилок ( $t$  менш).

**Теорема (границя Рейгера).** Кожний лінійний блоковий код, що виправляє всі пакети довжиною  $t$  і менш, повинен містити щонайменше  $2t$  перевірочних символів.

### Виправлення багаторазових помилок

Код Ріда-Соломона є одним з найбільш потужних кодів, що виправляють багаторазові пакети помилок. Застосовується в каналах, де пакети помилок можуть утворюватися настільки часто, що їх уже не можна виправляти за допомогою кодів, що виправляють одиночні помилки.  $(q^m - 1, q^m - 1 - 2t)$ -код Ріда-Соломона над полем  $GF(q^m)$  з кодовою відстанню  $d = 2t + 1$  можна розглядати як  $((q^m - 1)m, (q^m - 1 - 2t)m)$ -код над полем  $GF(q)$ , що може виправляти будь-яку комбінацію помилок, зосереджену в  $t$  або меншому числі блоків з  $m$  символів.

Найбільше число блоків довжини  $m$ , які може торкнути пакет довжини  $l_i$ , де  $l_i \leq mt_i - (m - 1)$ , не перевершує  $t_i$ , тому код, що може виправити  $t$  блоків помилок, завжди може виправити й будь-яку комбінацію з  $r$  пакетів загальної довжини  $l$ , якщо  $l + (m - 1) \leq mt$ .

### Практична реалізація

Кодування за допомогою коду Ріда-Соломона може бути реалізовано двома способами: систематичним і несистематичним.

При несистематичному кодуванні інформаційне слово множиться на якийсь полином, що неприводиться, у полі Галуа. Отримане закодоване слово повністю відрізняється від вихідного й для добування інформаційного слова потрібно виконати операцію декодування й уже потім можна перевірити дані на зміст помилок. Таке кодування вимагає більші витрати ресурсів тільки на добування інформаційних даних, при цьому вони можуть бути без помилок.

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

При систематичному кодуванні до інформаційного блоку з  $k$  символів приписуються  $2t$  перевірочних символів, при обчисленні кожного перевірочного символу використовуються всі  $k$  символів вихідного блоку.

У цьому випадку немає витрат ресурсів при добуванні вихідного блоку, якщо інформаційне слово не містить помилок, але кодер/декодер повинен виконати  $k(n - k)$  операцій додавання й множення для генерації перевірочних символів. Крім того, тому що всі операції проводяться в поле Галуа, те самі операції кодування/декодування вимагають багато ресурсів і часу. Швидкий алгоритм декодування, заснований на швидкому перетворенні Фур'є, виконується за час порядку  $\ln n^2$ .

### Кодування

При операції кодування інформаційний поліном множиться на багаточлен, що породжує. Множення вихідного слова  $S$  довжини  $k$  на не приводиться поліном, що, при систематичному кодуванні можна виконати в такий спосіб:

- До вихідного слова приписуються  $2t$  нулів, виходить поліном  $T = Sx^{2t}$ .
- Цей поліном ділиться на поліном, що породжує,  $G$ , перебуває залишок  $R$ ,  $Sx^{2t} = QG + R$ , де  $Q$  – частка.
- Цей залишок й буде коригувальним кодом Ріда-Соломона, він приписується до вихідного блоку символів. Отримане кодове слово  $C = Sx^{2t} + R$ .

Кодер будується зі регістрів зсуву, суматорів і перемножувачів. Регістр зсуву складається з комірок пам'яті, у кожній з яких перебуває один елемент поля Галуа.

Наведений як приклад кодер Ріда-Соломона генерує 16 коригувальних байт, що дозволяє виправляти до 8 і виявляти до 16 помилок у кадрі даних.

Перемножувачі на константи  $GF(0)...GF(15)$  у полі Галуа реалізуються в такий спосіб: спочатку вихідне число й константа перетворюються в індексну форму, потім складаються в межах байта без обліку переносу.

Результатом операції є результат додавання, перетворена обернено в поліноміальну форму.

					ВКРМ-123.25.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

При переході від однієї форми подання даних до іншої доцільно використовувати таблицю істинності розміром 256 байт, що становить ємність одного ЕАВ (Embedded Array Block – блок зосередженої пам'яті). Для реалізації кодера потрібно 16 таких перемножувачів, при цьому те саме число множиться на різні константи, що дозволяє використовувати для його перекладу в індексну форму один ЕАВ.

Для перекладу результатів у поліноміальну форму потрібно вже 16 таких таблиць, що вимагає застосування ІМС FPGA дуже великої ємності. У запропонованій схемі використовується тактування кодера із частотою, в 8 разів перевищуючу частоту надходження байт даних.

Це дає можливість використовувати дві пари «суматор – ЕАВ», мультиплексує константи на входах суматорів і дозволяє роботу регістрів-накопичувачів у моменти появи відповідних даних на виходах засувок ЕАВ.

На структурній схемі кодера (рисунок 3.2) символу «С» відповідають дві константи  $GF(n)$ .

Символ «L» у логіці регістрів-накопичувачів відповідає наступний: вихід компаратора нуля (символи CMP0 і SYNC) дозволяє роботу схеми «АБО що виключає », на входи якої подаються вихід попереднього регістра й ЕАВ. Якщо ж вектор зворотного зв'язку дорівнює "0", схема пропускає дані з виходу попереднього регістра-накопичувача на вхід наступного.

У результаті кодер з урахуванням схеми синхронізації (на рисунку не показана) займає 255 LE (Logic Element – логічний елемент) і 3 ЕАВ, що дозволяє розмістити його в ІМС EPF10K10. Після оптимізації розміщення схеми на кристалі FPGA швидкодія схеми досяглася 11,57 МГц (частота надходження байт даних, далі – байтова частота).

При використанні ІМС EPF10K20, у складі якої 6 ЕАВ, використовуючи 4 пари "суматор – ЕАВ", можна тактувати кодер із частотами, що перевищують байтову частоту не в 8, а в 4 рази, що дозволить підняти її до 25...30 МГц.

## Декодування

Декодер, що працює по авторегресивному спектральному методі декодування, послідовно виконує наступні дії:

- Обчислює синдром помилки.
- Будує поліном помилки.
- Знаходить корінь даного полінома.
- Визначає характер помилки.
- Виправляє помилки.

## Обчислення синдрому помилки

Обчислення синдрому помилки виконується синдромним декодером, що ділить кодове слово на багаточлен, що породжує. Якщо при діленні виникає остача, то в слові є помилка. Остача від ділення є синдромом помилки.

## Побудова полінома помилки

Обчислений синдром помилки не вказує на положення помилок. Ступінь полінома синдрому дорівнює  $2t$ , що багато менше ступеня кодового слова  $n$ . Для одержання відповідності між помилкою і її положенням у повідомленні будується поліном помилок.

Поліном помилок реалізується за допомогою алгоритму Берлекемпа-Мессі, або за допомогою алгоритму Евкліда. Алгоритм Евкліда має просту реалізацію, але вимагає більших витрат ресурсів. Тому частіше застосовується більше складний, але менш затратоємний алгоритм Берлекемпа-Мессі. Коефіцієнти знайденого полінома безпосередньо відповідають коефіцієнтам помилкових символів у кодовому слові.

Алгоритм Евкліда виконується наступним чином.

Нехай  $a$  і  $b$  суть цілі числа, не рівні одночасно нулю, і послідовність чисел  $a, b, r_1 > r_2 > r_3 > r_4 > \dots > r_n$  визначена тим, що кожне  $r_k$  це остача від ділення перед-попереднього числа на попереднє, а передостаннє ділиться на останнє націло, тобто

$$a = bq_0 + r_1$$

$$\begin{aligned}
 b &= r_1q_1 + r_2 \\
 r_1 &= r_2q_2 + r_3 \\
 &\dots \\
 r_{n-1} &= r_nq_n
 \end{aligned}
 \tag{3.3}$$

Тоді  $(a,b)$ , найбільший загальний дільник  $a$  і  $b$ , дорівнює  $r_n$ , останньому ненульовому члену цієї послідовності.

Існування таких  $r_1, r_2, \dots$ , тобто можливість ділення з остачею  $m$  на  $n$  для будь-якого цілого  $m$  і цілого  $n \neq 0$ , доводиться індукцією по  $m$ .

Коректність цього алгоритму випливає з наступних двох тверджень:

- Нехай  $a = bq + r$ , тоді  $(a,b) = (b,r)$ .
- $(0,r) = r$ . для будь-якого ненульового  $r$ .

### Знаходження корня

На цьому етапі шукаються коріння полінома помилки, що визначають положення перекручених символів у кодовому слові. Реалізується за допомогою процедури Ченя, рівносильній повному перебору. У поліном помилок послідовно підставляються всі можливі значення, коли поліном звертається в нуль – коріння знайдені.

### Визначення характеру помилки і її виправлення

По синдрому помилки й знайдених корінь полінома за допомогою алгоритму Форни визначається характер помилки й будується маска перекручених символів. Ця маска накладається на кодове слово за допомогою операції XOR і перекручені символи відновлюються. Після цього відкидаються перевірочні символи й виходить відновлене інформаційне слово.

Розглянувши усі блоки функціональної схеми перейдемо до розгляду діаграми взаємодії процесів, які відбуваються у системі.

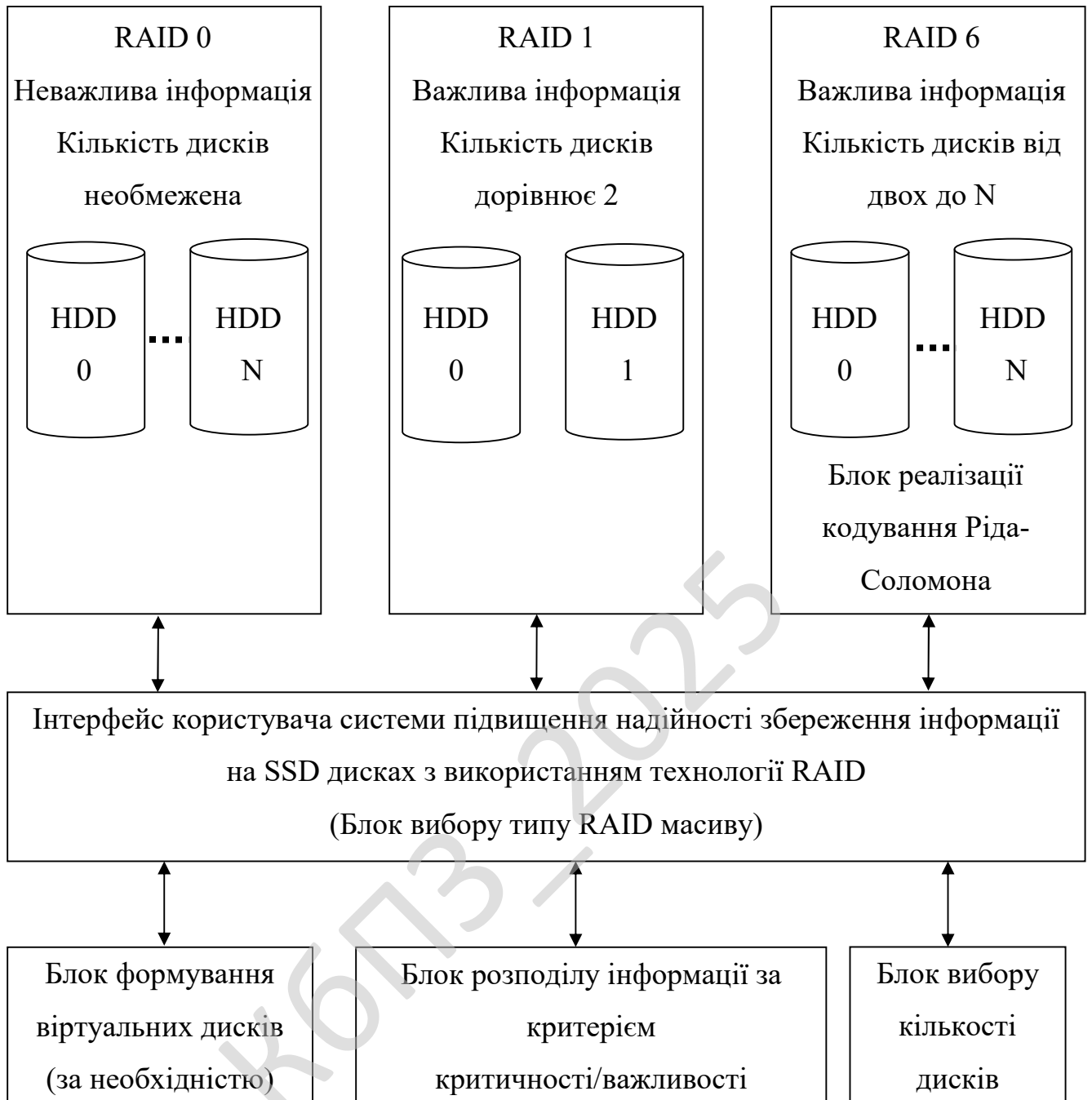


Рисунок 3.10 – Функціональна схема системи

Блок вибору типу RAID масиву дозволяє автоматично, в залежності від заданих параметрів у попередніх блоках, визначити, який саме тип RAID-масиву потрібен для зберігання тієї, або іншої інформації.

### 3.4 Розробка діаграми процесів

Діаграма процесів розробленої системи зображена на рисунку 3.11.

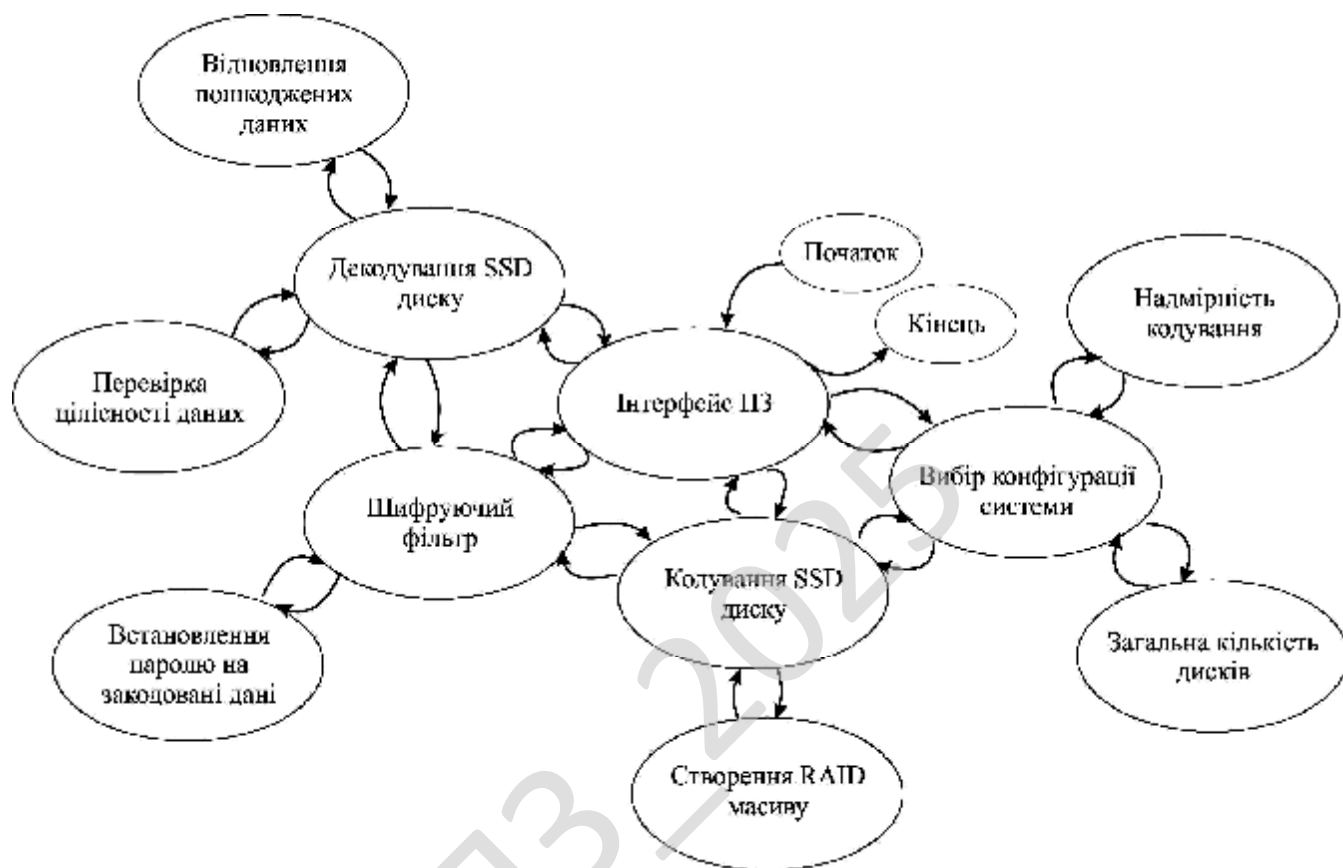


Рисунок 3.11 – Діаграма взаємодії процесів

Після початку роботи розробленого ПЗ ми потрапляємо до головного блоку системи звідки через ланку дій відбувається наступне:

- Інтерфейс ПЗ.
- Вибір конфігурації системи.
- Надмірність кодування.
- Загальна кількість дисків.
- Кодування SSD диску.
- Створення RAID масиву.
- Шифруючий фільтр.

- Встановлення паролю на закодовані дані.
- Декодування SSD диску.
- Відновлення пошкоджених даних.
- Перевірка цілісності даних.

Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі. Діаграма взаємодії процесів використовується для візуалізації процесів обробки даних (структурне проектування).

Для розробника вважається звичним спочатку креслити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи. Ця діаграма в подальшому підлягає уточненню шляхом деталізації процесів та потоків даних з метою показати систему що розробляється.

Таким чином, розглянувши опис системи, структурну, функціональну схеми системи, та діаграму взаємодії процесів перейдемо до опису блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

КБПЗ-2025

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>55</b>

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

### 4.1 Блок-схеми та опис алгоритмів функціонування системи

Первинною стадією без якої не відбувається розробка програмного забезпечення це звичайно розробка блок-схем.

На рисунку 4.1 зображена основна блок-схема програми, на рисунку 4.2 зображено роботу підпрограми. З якої 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 – перевірка поточного стану 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.

Блок-схеми є першоджерелами стратегії розвитку ПЗ. Тому від точності і детальної блок-схеми залежить результат всієї програми.

При виборі початкової точки відліку при побудові схем я враховував, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об'єктно-орієнтована що вимагає оптимізації програми високого рівня, також те, що при розробці програми слід надати особливу увагу модулю підвищення надійності збереження інформації.

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.

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>56</b>

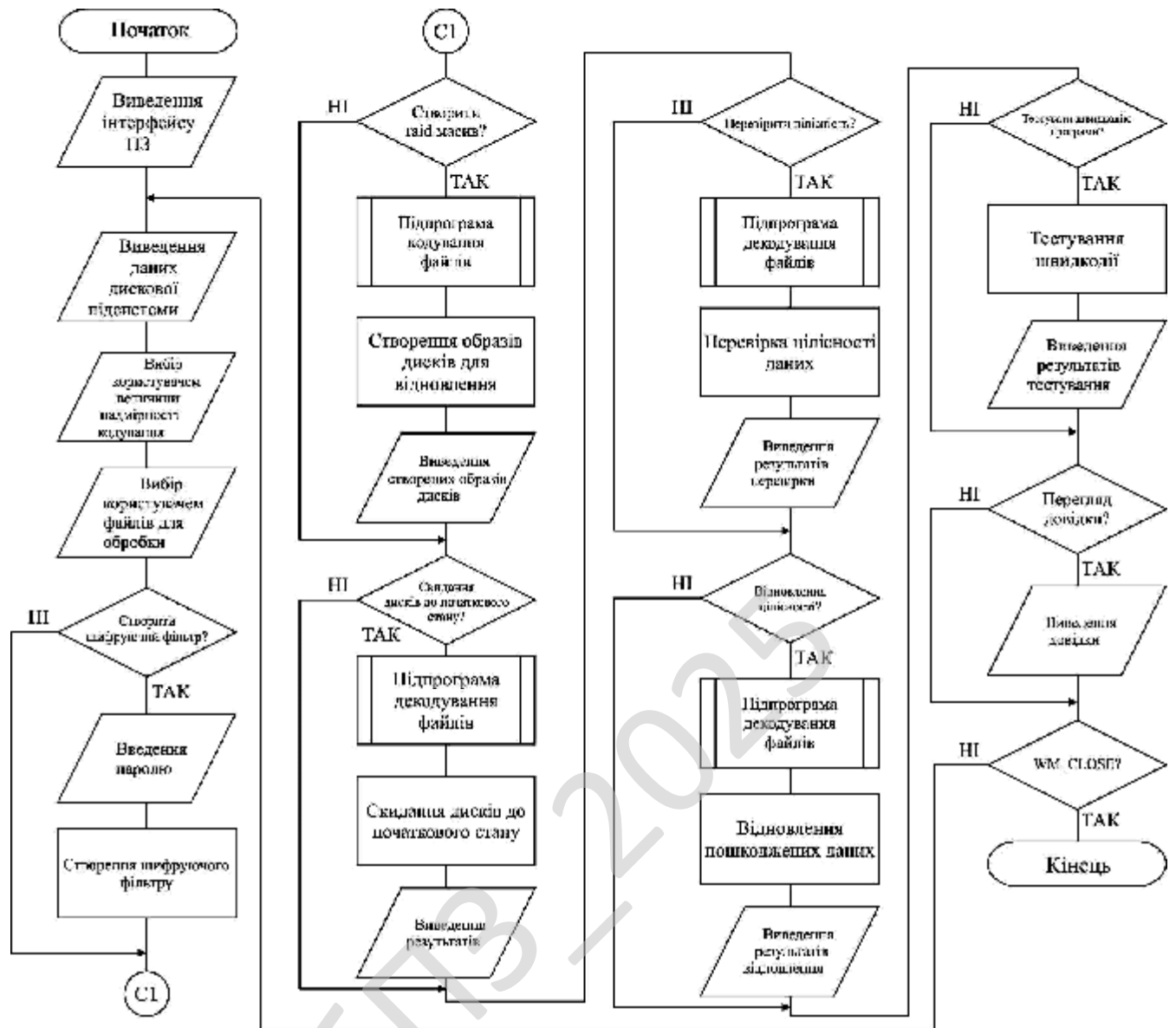


Рисунок 4.1 – Блок схема основної програми

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.

UML може бути застосовано на всіх етапах життєвого циклу аналізу бізнес-систем і розробки прикладних програм.

Різні види діаграм які підтримуються UML, і найбагатший набір можливостей представлення певних аспектів системи робить UML універсальним засобом опису як програмних, так і ділових систем.

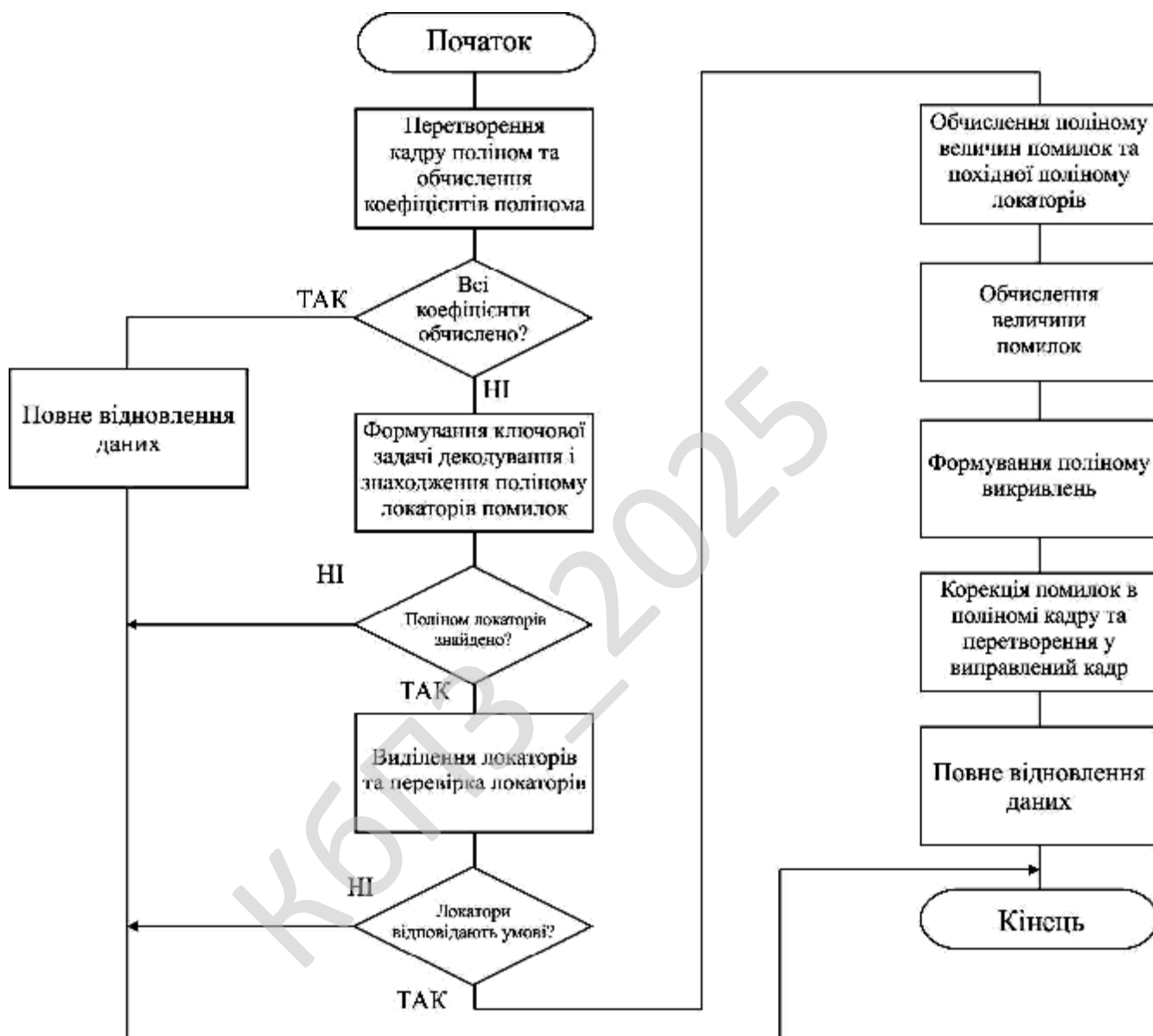


Рисунок 4.2 – Блок схема підпрограми

Діаграми дають можливість представити систему (як ділову, так і програмну) у такому вигляді, щоб її можна було легко перевести в програмний код. Основною причиною використання мови UML є спілкування розробників між собою.

Крім того, UML спеціально створювалася для оптимізації процесу розробки програмних систем, що дозволяє збільшити ефективність їх реалізації у кілька разів і помітно поліпшити якість кінцевого продукту.

UML прекрасно зарекомендувала себе в багатьох успішних програмних проектах. Засоби автоматичної генерації кодів дозволяють перетворювати моделі мовою UML у вихідний код об'єктно-орієнтованих мов програмування, що ще більш прискорює процес розробки.

Практично усі CASE-засоби (програми автоматизації процесу аналізу і проектування) мають підтримку UML. Моделі розроблені в UML, дозволяють значно спростити процес кодування і направити зусилля програмістів безпосередньо на реалізацію системи.

Діаграми підвищують супроводжуваність проекту і полегшують розробку документації.

UML необхідний:

- Керівникам проектів, які керують розподілом завдань і контролем за проектом.
- Проектувальникам інформаційних систем які розробляють технічні завдання для програмістів.
- Бізнес-аналітикам, які досліджують реальну систему і здійснюють інжиніринг і реінжиніринг бізнесу компанії.
- Програмістам які реалізують модулі інформаційної системи.

При модифікації системи об'єктний підхід дозволяє легко включати в систему нові об'єкти і виключати застарілі без істотної зміни її життєздатності. Використання побудованої моделі при модифікаціях системи дає можливість усунути небажані наслідки змін, оскільки вони не ламають структури системи, а тільки змінюють поведінку об'єктів.

При складанні блок-схем програмного забезпечення і напрацювання алгоритмів я зіткнувся з масою проблем, які вимагали напрацювання процедур і

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

функцій над основною проблематикою. Для чого були створені додаткові класи, типи даних і константи, що забезпечило вирішення проблем.

Розглянемо опис алгоритмів функціонування системи. Нижче наведемо ту частину кода, яка виконує вищеперераховані дії. Використаний кодер Ріда-Соломона. Кодуємі дані передаються через масив  $data[i]$ , де  $i = 0 \dots (k-1)$ , а згенеровані символи парності заносяться в масив  $b[0] \dots b[2t-1]$ . Вихідні й результуючі дані повинні бути представлені в поліноміальній формі (тобто у звичайній формі машинного подання даних).

Кодування виражається з використанням зсувного feedback-регістра, заповненого відповідними елементами масиву  $g[]$  з породженням поліномом усередині, процедура генерації якого вже обговорювалася в попередньому розділі. Згенероване кодове слово описується наступною формулою:

$$C(x) = data(x) * x^{(n-k)} + b(x),$$

де  $^{\wedge}$  означає зведення в ступінь.

```
encode_rs()
{
    int i, j;
    int feedback;
    // ініціалізація поля біт парності нулями
    for (i = 0; i < n - k; i++) b[i] = 0;
    // обробляємо всі символи
    // вихідних даних праворуч ліворуч
    for (i = k - 1; i >= 0; i--)
    {
        // готовимо (data[i] + b[n - k - 1]) до множення на g[i]
        // тобто складаємо черговий "захоплений" символ вихідних
        // даних з молодшим символом біт парності (відповідного
        // "регістру" b2 t-1, і переводимо його в
        // індексну форму, зберігаючи результат у регістрі feedback;
        // як ми вже говорили, сума двох індексів є добуток поліномів
        feedback = index_of[data[i] ^ b[n - k - 1]];
        // є ще символи для обробки?
        if (feedback != -1)
        {
            // здійснюємо зрушення ланцюга bx-регістрів
```

```

        for (j = n - k - 1; j > 0; j--){
// якщо поточний коефіцієнт g - це дійсний
// (тобто ненульовий коефіцієнт, те
// множимо feedback на відповідний g-коефіцієнт
// і складаємо його з наступних елементів ланцюжка
            if (g[j] != -1) b[j] = b[j - 1] ^ alpha_to[(g[j] +
feedback) % n];
        else
// якщо поточний коефіцієнт g - це нульовий коефіцієнт,
// виконуємо одне лише зрушення без множення, переміщаючи
// символ з одного m-регістра в інший
            b[j] = b[j - 1];
// закріплюємо вихідний символ у крайній лівий b 0-регістр
            b[0] = alpha_to[(g[0] + feedback) % n];
        }
        else
        {
// розподіл завершений,
// здійснюємо останнє зрушення регістра,
// на виході регістра буде частка, що губиться,
// а в самому регістрі - шуканий залишок
            for (j = n - k - 1; j > 0; j--){ b[j] = b[j - 1]; b[0] = 0;
            }
        }
    }
}

```

Розглянувши алгоритм та програмну реалізацію процесу кодування перейдемо до процесу декодування закодованого тексту.

Декодування кодів Ріда-Соломона являє собою досить складне завдання, рішення якого виливається в громіздкий, заплутаний і надзвичайно ненаглядний програмний код, що вимагає від розроблювача великих знань у багатьох областях вищої математики.

Типова схема декодування, що одержала назву авторегресійного спектрального методу декодування, складається з наступних кроків:

- Обчислення синдрому помилки (синдромний декодер).
- Побудови полінома помилки, здійснювана або за допомогою високоефективного, але складно реалізованого алгоритму Берлекемпа-Мессі, або за допомогою простого, але гальмового Евклідового алгоритму.

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>61</b>

– Знаходження корінь даного полінома, що звичайно вирішується лобовим перебором.

– Визначення характеру помилки, що зводиться до побудови бітової маски, що обчислюється на основі обігу алгоритму Форни або будь-якого іншого алгоритму обігу матриці.

– Нарешті, виправлення помилкових символів, шляхом накладення бітової маски на інформаційне слово й послідовне інвертування всіх перекручених біт через операцію XOR.

Наведемо вихідний текст підпрограми декодера Ріда-Соломона. Процедура декодування кодів Ріда-Соломона складається з декількох кроків: спочатку ми обчислюємо  $2t$ -символьний синдром шляхом постановки  $\alpha^{*i}$  в  $\text{recd}(x)$ , де  $\text{recd}$  – отримане кодове слово, попередньо переведене в індексну форму.

По факту обчислення  $\text{recd}(x)$  ми записуємо черговий символ синдрому в  $s[i]$ , де  $i$  приймає значення від 1 до  $2t$ , залишаючи  $s[0]$  рівним нулю. Потім, використовуючи ітеративний алгоритм Берлекемпа (Berlekamp), ми знаходимо поліном локатора помилки –  $\text{elp}[i]$ .

Якщо ступінь  $\text{elp}$  перевищує собою величину  $t$ , ми неспроможні скорегувати всі помилки й обмежуємося виводом повідомлення про непереборну помилку, після чого робимо аварійний вихід з декодера. Якщо ж ступінь  $\text{elp}$  не перевищує  $t$ , ми підставляємо  $\alpha^{*i}$ , де  $i = 1 \dots n$  в  $\text{elp}$  для обчислення корінь полінома. Обіг знайдений корінь дає нам позиції перекручених символів.

Якщо кількість певних позицій перекручених символів менше ступеня  $\text{elp}$ , перекручуванню піддалося більш ніж  $t$  символів і ми не можемо відновити їх. У всіх інших випадках відновлення оригінального вмісту перекручених символів цілком можливо.

У випадку, коли кількість помилок свідомо велика, для їхнього виправлення декодуємо символи проходять крізь декодер без яких або змін.

```
decode_rs()  
{  
    int i, j, u, q;
```







```

// корекція помилок можлива
// переводимо elp в індексну форму
    for (i = 0; i <= l[u]; i++) elp[u][i] = index_of[elp[u][i]];
// знаходження корінь полінома локатора помилки
//-----
    for (i = 1; i <= l[u]; i++) reg[i] = elp[u][i]; count = 0;
    for (i = 1; i <= n; i++)
    {
        q = 1 ;
        for (j = 1; j <= l[u]; j++)
            if (reg[j] != -1)
            {
                reg[j] = (reg[j] + j) % n;
                q ^= alpha_to[reg[j]];
            }
        if (!q)
        {
// записуємо корінь і індекс позиції помилки
            root[count] = i;
            loc[count] = n - i;
            count++;
        }
    }
    if (count == l[u])
    {
// немає корінь - ступінь elp < t помилок
// формуємо поліном z(x)
        for (i = 1; i <= l[u]; i++)
// Z[0] завжди дорівнює 1
        {
            if ((s[i] != -1) && (elp[u][i] != -1))
                z[i] = alpha_to[s[i]] ^ alpha_to[elp[u][i]];
            else
                if ((s[i] != -1) && (elp[u][i] == -1))
                    z[i] = alpha_to[s[i]];
                else
                    if ((s[i] == -1) && (elp[u][i] != -1))
                        z[i] = alpha_to[elp[u][i]];
                    else
                        z[i] = 0 ;

            for (j = 1; j < i; j++)
                if ((s[j] != -1) && (elp[u][i - j] != -1))

```

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

```

                                z[i] ^= alpha_to[(elp[u][i - j] + s[j]) % n];
// переводимо z[i] в індексну форму
                                z[i] = index_of[z[i]];
                                }
// обчислення значення помилок у позиціях loc[i]
//-----
                                for (i = 0; i < n; i++)
                                {
                                        err[i] = 0;
// переводимо recd[] у поліноміальну форму
                                        if (recd[i] != -1) recd[i] = alpha_to[recd[i]]; else recd[i] = 0;
                                }
// спочатку обчислюємо чисельник помилки
                                for (i = 0; i < l[u]; i++)
                                {
                                        err[loc[i]] = 1;
                                        for (j = 1; j <= l[u]; j++)
                                                if (z[j] != -1)
                                                        err[loc[i]] ^= alpha_to[(z[j] + j * root[i]) % n];
                                        if (err[loc[i]] != 0)
                                        {
                                                err[loc[i]] = index_of[err[loc[i]]];
                                                q = 0 ;
// формуємо знаменник коефіцієнта помилки
                                                for (j = 0; j < l[u]; j++)
                                                        if (j != i)
                                                                q += index_of[1 ^ alpha_to[(loc[j] + root[i]) % n]];
                                                q = q % n; err[loc[i]] = alpha_to[(err[loc[i]] - q + n) % n];
// recd[i] повинен бути в поліноміальній формі
                                                recd[loc[i]] ^= err[loc[i]];
                                        }
                                }
                                }
                                else
// немає корінь,
// рішення системи рівнянь неможливо, тому що ступінь elp >= t
                                {
// переводимо recd[] у поліноміальну форму
                                        for (i = 0; i < n; i++)
                                                if (recd[i] != -1) recd[i] = alpha_to[recd[i]];
                                        else
                                                recd[i] = 0;

```



версія алгоритму Srypton v0.5. Однак, як казав Чьо Лім Хун, йому не вистачало часу для розробки повної версії. І вже на першому етапі конкурсу AES в ході аналізу алгоритмів, версія Srypton v0.5 була замінена на версію Srypton v1.0. Відмінність нової версії від первинної полягала в зміні таблиці замін та в модифікації процесу розширення ключа.

Як і інші учасники конкурсу AES, Srypton призначений для шифрування 128-бітових блоків даних[2]. При шифруванні використовуються ключі шифрування для декількох фіксованих розмірів – від 0 до 256 біт з кратністю 8 бітів. Структура алгоритму Srypton – структура «Квадрата» – багато в чому схожа на структуру алгоритму Square, створеного в 1997 році. Криптографічні перетворення для алгоритмів з даною структурою можуть бути виконані як для цілих рядків і стовпців масиву, так і над окремими його байтами. (Варто зазначити, що алгоритм Square був розроблений авторами майбутнього переможця конкурсу AES – авторами алгоритму Rijndael – Вінсентом Ріджменом і Джоан Дейменом.)

### **Шифрування**

Алгоритм Srypton являє 128-бітовий блок шифруємих даних у вигляді байтового масиву  $4 \times 4$ , над якими в процесі шифрування проводиться кілька раундів перетворень. У кожному раунді передбачається послідовне виконання наступних операцій:

- Таблична заміна  $\gamma$ ;
- Лінійне перетворення  $\pi$ ;
- Байтова перестановка  $\tau$ ;
- Операція  $\sigma$ .

### **Таблична заміна $\gamma$**

Алгоритм Srypton використовує 4 таблиці замін. Кожна з яких заміщає 8-бітне вхідне значення на вихідне такого ж розміру.

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

### **Лінійне перетворення $\pi$**

Тут використовується 4 спеціальні константи. Ці константи об'єднані в маскуючі послідовності

### **Байтова перестановка $\tau$**

Дана перестановка перетворює найпростішим чином рядок даних у стовпець.

### **Операція $\sigma$**

Дана операція є побітовим складанням всього масиву даних з ключем раунду. Зауважимо, саме 12 раундів шифрування рекомендується автором алгоритму Чьо Хун Лімом, проте сувора кількість раундів не встановлена.

КБПЗ\_2025

					VKPM-123.25.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

## 5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розглянемо розроблене ПЗ яке зображено на рисунку 5.1. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні розділи:

- Файл.
- Інструменти.
- Параметри.
- Довідка.

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.

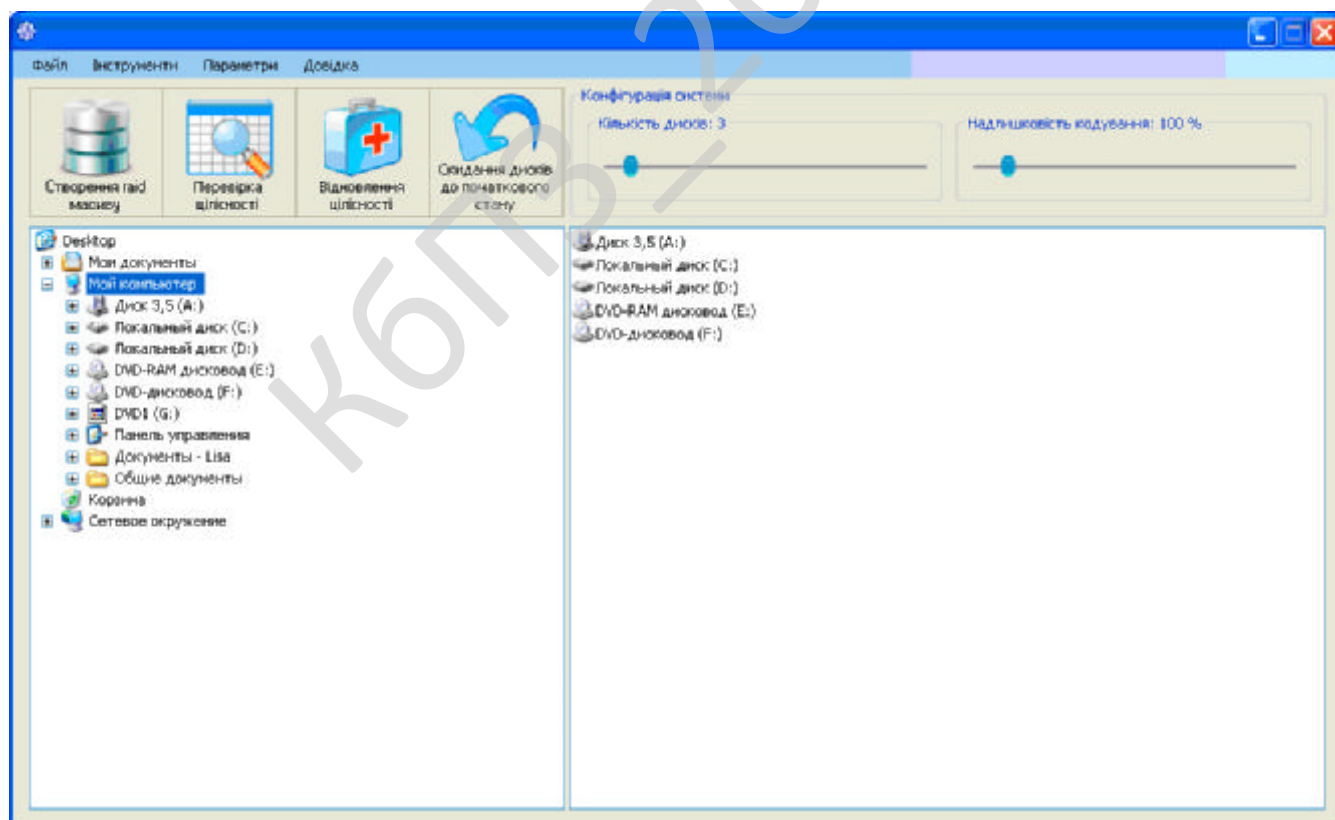


Рисунок 5.1 – Основне вікно програми

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

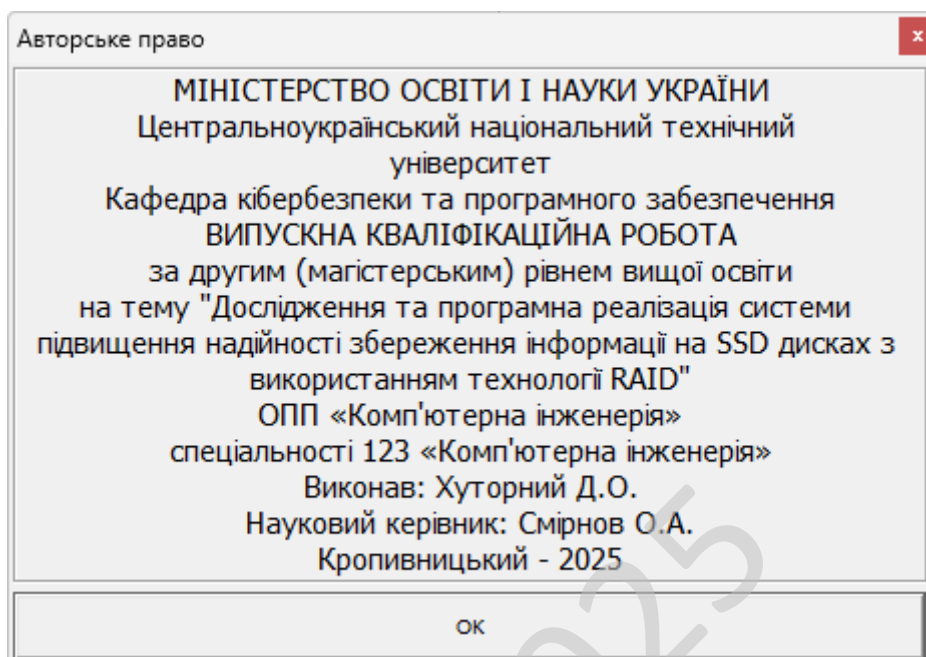


Рисунок 5.2 – Авторське право

Обрано умови розповсюдження – Shareware. Під умовно-безплатним програмним забезпеченням можна розуміти спосіб або метод розповсюдження комерційного ПЗ на ринку (тобто на шляху до кінцевого користувача), при якому випробувачеві пропонується обмежена за можливостями (не повнофункціональна або демонстраційна версія), терміном дії (тріал версія) або версія з вбудованим набридливим нагадуванням про необхідність оплати використання програми. В угоді про використання (ліцензії для кінцевого користувача, EULA) також може бути обумовлена заборона на комерційне або професійне (не тестове) її використання. Основний принцип умовно-безплатного ПЗ – «спробуй, перш ніж купити» (try before you buy). ПЗ що поширюється як умовно-безплатний, надається користувачам безоплатно. Звичайно користувач платить тільки за час завантаження файлів через Інтернет або за носій (CD диск, флешку, ключ). Протягом певного терміну, що становить зазвичай тридцять днів, він може

					ВКРМ-123.25.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

користуватися програмою, тестувати її, освоювати її можливості. Якщо після закінчення цього терміну користувач вирішить продовжити використання ПЗ, він зобов'язаний купити його (zareєstrуватися), заплативши авторові певну суму.

КБПЗ - 2025

					VKPM-123.25.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

## 6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи підвищення надійності збереження інформації на SSD дисках з використанням технології RAID.

*Метою розробки є дослідження та програмна реалізація системи підвищення надійності збереження інформації на SSD дисках з використанням технології RAID.*

*Об'єктом дослідження є процес підвищення надійності збереження інформації на SSD дисках з використанням технології RAID.*

*Предметом дослідження є методи підвищення надійності збереження інформації на SSD дисках з використанням технології RAID.*

*Методи дослідження базуються на методах теорії зберігання даних, методах математичної статистики, методах розробки програмного забезпечення.*

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

– Удосконалено метод підвищення надійності збереження інформації на SSD дисках з використанням технології RAID.

– Розроблено вітчизняний продукт підвищення надійності збереження інформації на SSD дисках з використанням технології RAID, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-123.25.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

## 7 МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ІТ-ПРОЄКТУ

### 7.1 Визначення цільової аудиторії кінцевого готового продукту

Результати дослідження та розробки системи мережевого моніторингу та контролю стану ІТ можуть бути насамперед корисними для підприємств, які мають розгалужену ІТ-інфраструктуру й використовують сервери, мережеве обладнання та корпоративні сервіси для підтримки своєї діяльності. Для таких компаній стабільність і безперебійність роботи інформаційних систем є критично важливими, тому можливість своєчасного виявлення несправностей або перевантажень стає суттєвою конкурентною перевагою. Саме система моніторингу допомагає контролювати роботу мережевих пристроїв у режимі реального часу, виявляючи проблеми ще до того, як вони вплинуть на користувачів.

Особливий інтерес до таких систем можуть проявити ІТ-компанії, які займаються наданням послуг хостингу, розробкою програмного забезпечення або підтримкою клієнтів. Для них швидкість реагування на інциденти та якість технічного обслуговування є показниками репутації, а отже, від роботи системи моніторингу залежить рівень довіри клієнтів і лояльність користувачів. Такі підприємства часто працюють у середовищі, де навіть хвилинна затримка чи зупинка сервера призводить до фінансових збитків, тому автоматизація контролю за станом мережі – це не розкіш, а необхідність.

Крім комерційних компаній, результати дослідження будуть актуальними для державних структур, освітніх установ і організацій, які мають внутрішні мережі та зберігають великі обсяги інформації. У таких установах впровадження системи моніторингу підвищує ефективність роботи ІТ-відділів, зменшує ризик втрати даних і допомагає раціонально використовувати наявні ресурси.

					ВКРМ-123.25.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

Не менш важливим є значення цієї розробки для навчальних і наукових закладів. Вони можуть використовувати систему як навчальну платформу для підготовки фахівців у сфері інформаційних технологій. Студенти отримують можливість не лише спостерігати за реальною роботою системи моніторингу, а й аналізувати дані, моделювати різні ситуації та вчитися реагувати на інциденти. Таким чином, результати дослідження мають універсальний характер і можуть бути впроваджені як у бізнесі, так і в освіті.

## 7.2 Оцінка привабливості шляхом застосування методів експертних оцінок

Для оцінки привабливості програмного продукту було проведено експертне опитування серед фахівців у галузі IT-інфраструктури, адміністраторів систем і представників компаній, що мають досвід використання схожих рішень. Експертам було запропоновано оцінити систему за основними критеріями – функціональні можливості, надійність, простота впровадження, масштабованість, вартість експлуатації та потенційна економічна ефективність.

Більшість експертів високо оцінили саме інтелектуальну частину системи – можливість автоматичного сповіщення про інциденти, генерацію аналітичних звітів і прогнозування потенційних відмов обладнання. Особливо було відзначено, що система працює стабільно навіть при великому навантаженні й може адаптуватися до різних типів мережевої інфраструктури, що робить її універсальною.

За результатами оцінки середній рівень привабливості продукту склав 8,7 бала з 10 можливих. Експерти зазначили, що така система може мати великий попит серед середніх і великих підприємств, особливо якщо її вартість залишатиметься конкурентною. Також було підкреслено, що простота інтерфейсу та можливість кастомізації під конкретного користувача є суттєвими перевагами, які підвищують комерційний потенціал рішення.

					ВКРМ-123.25.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

Таким чином, метод експертних оцінок показав, що система має високу ринкову привабливість, відповідає актуальним потребам бізнесу та може стати успішним продуктом за умови належного маркетингового просування та підтримки користувачів.

### 7.3 Вибір методу оцінки вартості ПЗ

Для оцінки вартості розробки системи мережевого моніторингу та контролю стану ІТ доцільно використовувати витратний метод. Він передбачає визначення всіх фактичних витрат, які були понесені під час створення програмного продукту, включаючи оплату праці розробників, витрати на апаратне забезпечення, ліцензії, тестування та впровадження. Такий підхід дозволяє точно визначити базову собівартість проєкту, що є особливо важливим для невеликих команд і стартапів.

Однак, у випадку комерційного впровадження, доцільно поєднати цей підхід із дохідним методом. Дохідний метод дає змогу оцінити майбутні вигоди, які підприємство отримає після впровадження системи. Наприклад, скорочення простоїв серверів, підвищення ефективності роботи персоналу та зменшення витрат на ручну діагностику мережі є прямими джерелами економічної вигоди.

Такий комбінований підхід дозволяє не лише визначити початкову вартість розробки, а й обґрунтувати економічну доцільність проєкту. Він допомагає потенційним інвесторам побачити не просто витрати, а реальні фінансові перспективи, які відкриває впровадження системи.

У результаті використання комбінованої моделі оцінки можна отримати повну картину вартості та окупності проєкту, що стане основою для прийняття управлінських рішень щодо його реалізації чи масштабування.

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

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

Компанія має розгалужену ІТ-інфраструктуру, яка включає сервери, мережеве обладнання, робочі станції, системи зберігання даних і корпоративні сервіси. До впровадження системи моніторингу контроль за станом мережі здійснювався вручну: адміністратори виявляли проблеми лише після звернень користувачів або повного виходу сервісів із ладу. Це призводило до простоїв, затримок у роботі та фінансових втрат. Основна мета впровадження системи мережевого моніторингу – забезпечити цілодобове автоматичне відстеження стану обладнання, серверів і додатків, оперативне реагування на інциденти, зниження кількості простоїв і запобігання критичним збоєм у роботі ІТ-інфраструктури. Вхідні дані зафіксовано в таблиці 7.1.

Таблиця 7.1 – Вихідні дані для розрахунку

Показник	До впровадження	Після впровадження	Економічний ефект
Кількість простоїв серверів на рік	20 випадків	5 випадків	-15
Середня тривалість простою одного сервера	4 години	1 година	-3 години
Середні втрати підприємства за 1 годину простою	25 000 грн	5 000 грн	-20 000 грн
Витрати на ручну діагностику й усунення збоїв	300 000 грн/рік	150 000 грн/рік	-150 000 грн
Вартість впровадження системи моніторингу	—	—	450 000 грн
Річні витрати на підтримку системи	—	—	100 000 грн



Наступним етапом є розширення партнерських зв'язків. Доцільно співпрацювати з ІТ-компаніями, які займаються інтеграцією корпоративних систем, адже вони можуть пропонувати продукт своїм клієнтам як частину комплексного рішення. Водночас слід розробити гнучку цінову політику – наприклад, ліцензування за кількістю пристроїв або модель передплати, що зробить продукт доступнішим для малого та середнього бізнесу.

Просування має супроводжуватися технічною підтримкою користувачів, оновленнями та навчанням персоналу. Це створює позитивний досвід використання продукту та сприяє формуванню довгострокових відносин із клієнтами. У підсумку правильна стратегія просування допоможе не лише збільшити продажі, а й побудувати впізнаваний бренд на ринку ІТ-рішень.

## 7.6 Оптимізація каналів збуту та шляхів реалізації ПЗ

Для оптимізації каналів збуту варто поєднати прямі продажі з цифровими платформами розповсюдження програмного забезпечення. Власний сайт компанії може стати не лише вітриною продукту, а й каналом комунікації з клієнтами, де вони зможуть отримати демо-версію, консультацію або підтримку. Це сприятиме зниженню витрат на маркетинг і збільшенню довіри.

Додатково ефективним буде впровадження партнерської програми для системних інтеграторів і реселерів, які вже мають доступ до корпоративних клієнтів. Така модель дозволяє розширити охоплення ринку без суттєвих додаткових інвестицій.

Також можна запропонувати гібридну форму реалізації: ліцензування для великих компаній і модель SaaS (Software as a Service) для малого бізнесу. Це підвищить доступність системи та дозволить гнучко реагувати на потреби різних сегментів ринку.

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>80</b>

Ключовим напрямом оптимізації збуту є створення якісного сервісу після продажу – технічна підтримка, регулярні оновлення, аналітичні звіти. Усе це забезпечує стабільність роботи клієнта й стимулює його до подальшої співпраці.

### **7.7 Визначення ключових факторів успіху конкретного проєкту**

Основним фактором успіху є стабільність і надійність системи. Якщо система моніторингу працює без збоїв і забезпечує реальну користь, вона швидко здобуває довіру користувачів. Технологічна якість продукту, його здатність масштабуватися й інтегруватися з іншими ІТ-рішеннями відіграють ключову роль у його життєздатності.

Другим важливим чинником є професійна команда розробників і технічної підтримки. Клієнти цінують не лише продукт, а й можливість отримати швидко допомогу у випадку проблем або питань. Від рівня компетенції фахівців залежить не лише якість обслуговування, а й довгострокові відносини з партнерами.

Не менш значущим є гнучкість системи – можливість адаптувати її під специфіку кожного клієнта. Різні компанії мають різну інфраструктуру, тому універсальне, але налаштовуване рішення стає перевагою.

І, нарешті, успіх будь-якого ІТ-проєкту визначається здатністю постійно вдосконалюватися. Регулярні оновлення, впровадження нових технологій і зворотний зв'язок із користувачами формують довіру й підтримують актуальність продукту на ринку. Саме ці чинники разом створюють основу для стабільного розвитку та комерційного успіху системи моніторингу.

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>81</b>

## 8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

### 8.1 Вступ

Електронно-обчислювальна машина (ЕОМ) відіграє важливу роль у житті сучасної людини. Кожного дня мільйони людей використовують ЕОМ для пошуку необхідної інформації, спілкуванні у соціальних мережах, перегляду новин, роботи тощо. Багато людей користуються ЕОМ у професійних цілях, оскільки завдяки ЕОМ з'явилося багато нових професій. Тому для розробника хмарних сервісів так важливо розробити зручний інтерфейс для зручного сприйняття інформації, та необхідний функціонал, який буде відповідати необхідним вимогам та навантаженням. Все це вимагає багато часу та великого навантаження з боку розробників. Тому так важливо слідкувати за умовами праці, в яких відбувається робочий процес. Оскільки захворювання можуть бути спричинені надмірним фізичним або розумовим навантаженням, через велику нервово-емоційну напругу, або через виробниче середовище. В даному розділі магістерської роботи проведемо аналіз основних чинників при роботі програміста.

Законом України “Про охорону праці” регламентуються загальні положення державної політики в галузі охорони праці, а конкретизуються ці положення нормативно-правовими актами про охорону праці, зокрема Наказом Міністерства соціальної політики України 14.02.2018 № 207, який зареєстровано в Міністерстві юстиції України 25 квітня 2018 р. за №508/31960 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями», яким затверджено нормативно-правовий акт з охорони праці НПАОП 0.00-7.15-18, «Правила охорони праці під час експлуатації електронно-обчислювальних машин», та «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98.

					ВКРМ-123.25.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

## 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.

На роботу програміста впливають наступні фактори: невідповідний мікроклімат приміщення (температура, вологість), недостатня освітленість робочої зони, підвищений рівень шуму та електромагнітного випромінювання, порушення іонного складу повітря, неправильна ергономічна організація робочого місця, ризики, пов'язані із погіршенням зору, порушенням фізичного стану, стресом тощо.

Шкідливими факторами при роботі з персональним комп'ютером є неіонізуюче випромінювання промислової частоти, збільшене нервово- емоційне навантаження на оператора, збільшення навантаження на органи зору та дрібні стереостатичні рухи кінцівок. Ці фактори можуть викликати у працівника певні розлади здоров'я, зокрема підвищення артеріального тиску, кон'юктивіти, тендовагініти та інші захворювання.

Комп'ютер, як і будь-який електричний прилад, особливо при його неправильному підключенні, може бути джерелом ураження оператора електричним струмом. Саме тому всі працівники, які працюють з персональним комп'ютером, повинні мати першу (або другу) групу допуску з електробезпеки.

Через наявність зазначених факторів працівники, які працюють з персональними комп'ютерами, підлягають попередньому та періодичному медичному огляду згідно з пунктом 6.2.3 додатку 4 до наказу Міністерства охорони здоров'я України «Про затвердження Порядку проведення медичних оглядів працівників певних категорій» від 21 травня 2007 року № 246.

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83

### 8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста

Оптимальна температура в приміщенні для праці має становити 20-24°C, відносна вологість – 40-60 %, атмосферний тиск – 750 мм. рт. ст., запиленість не повинна перевищувати 10 мг/м<sup>3</sup>, швидкість руху повітря – 0,1 м/с.

Через те, що обчислювальна техніка є джерелом тепловиділення, організація мікроклімату потребує додаткових зусиль: кондиціонування, провітрювання, використання систем опалення тощо. Об'єм приміщень повинен передбачатися з урахуванням як мінімум 20 м<sup>3</sup> /на особу [4].

Монітори комп'ютерів є джерелом випромінювання, яке може зашкодити здоров'ю людини. Для забезпечення роботи з комп'ютером відстань від монітора повинна становити не менше 50 см, бажано використовувати монітори зі зниженим рівнем, скорочувати час безперервної роботи за комп'ютером (робити п'ятнадцяти хвилинні перерви після кожних півтори години праці). Також в приміщенні необхідно встановлювати іонізатори повітря, використовувати нейтралізатори та зволожувачі.

Комп'ютери та периферійні пристрої є джерелами шуму, висока інтенсивність якого може призвести до проблем з органами слуху та негативно впливати на психологічний стан. Рівень шуму на робочому місці не повинен перевищувати 50 дБА [5]. Для зменшення рівня шуму можна використовувати звукопоглинальні пристрої, а стіни приміщень з комп'ютерами можуть бути покриті звукопоглинальними матеріалами. Поряд із шумом часто виникає вібрація. Для зменшення рівня вібрації в приміщенні на поверхні необхідно встановлювати віброізолятори.

Ергономічні показники робочого місця програміста мають бути наступними: висота робочої поверхні повинна складати 720 мм, розмір поверхні має становити 1600 x 1000 мм; під столом повинен бути простір з розмірами по глибині 650 мм; стіл повинен мати підставку для ніг, розташовану під кутом

15° до поверхні; відстань клавіатури від краю столу має бути не більше 300 мм; відстань між очима й екраном повинна складати 40 – 80 см; стілець повинен мати підйомно-поворотний механізм; висота сидіння має регулюватися в межах 400 – 500 мм, глибина – не менше 380 мм, а ширина – не менше 400 мм, висота опорної поверхні спинки має бути не менше 300 мм, ширина – не менше 380 мм. Кут нахилу спинки стільця до площини сидіння повинен змінюватися в межах 90 – 110° [6].

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

В літню пору застосовується кондиціонер.

Для боротьби з пилом робляться регулярні провітрювання та вологі прибирання приміщенні.

У приміщенні знаходяться наступні джерела шуму: принтер Prinics PicKit M1 Smartphone Photo Printer White, електродвигуни вентиляторів ЕОМ.

Робота програміста передбачає постійний візуальний контакт з моніторами комп'ютерів, та, як наслідок, значне навантаження на зір. Традиційно, це зорова робота високої або середньої точності. Для зорової роботи високої точності загальне освітлення (розподіл світла у всьому об'ємі приміщення) має становити 300 лк, комбіноване освітлення (поєднання загального і місцевого освітлення) – 750 лк. Штучне освітлення повинно бути рівномірним та використовуватися в світлий і темний час доби. Джерелами штучного освітлення можуть слугувати люмінесцентні лампи. Правильне освітлення передбачає уникнення відблисків на екранах.

З 2019 року діють Державні будівельні норми України “Природне і штучне освітлення” – ДБН В.2.5-28:2018 [4], у яких прописані вимоги до використання всіх освітлювальних приладів, у т.ч. світлодіодних.

Працю працівника, який постійно працює за комп'ютером, згідно ДБН В.2.5-28:2018 [4], можна віднести до роботи з малою точністю (найменший

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

розмір об'єкта розрізнення від 1 до 5 мм) V-го розряду зорової роботи, з великою контрастністю об'єкта розрізнення (символів на екрані дисплея), з темним тлом (під розряд зорової роботи B). Приміщення можна віднести до 1-ої групи приміщень, у яких проводиться розрізнення об'єктів зорової роботи при фіксованому напрямку лінії зору того, що працює на робочу поверхню. Для такого типу приміщень і розряду зорової роботи нормоване значення коефіцієнта природної освітленості (КПО) робочої поверхні (при поєднаному, спільному освітленні), повинен становити не більше 1,5%, освітленість при штучному висвітленні повинна становити 300 Лк. [4], Крім того все поле зору повинно бути освітлено достатньо рівномірно – це основна гігієнічна вимога. Оскільки яскраве 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

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		86

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.

## 8.5 Розрахункова частина

Проведемо розрахунок штучного освітлення за методом коефіцієнта використання світлового потоку для приміщення ширина якого складає 6 м, довжина – 7 м, висота – 2,9 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  $S = 6 \times 7 = 42$  м<sup>2</sup>);

$K$  – коефіцієнт запасу, що враховує зменшення світлового потоку лампи в результаті забруднення світильників у процесі експлуатації (його значення

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		87

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, у нашому випадку  $Z = 1,1$ );

$n$  – коефіцієнт використання світлового потоку, (відношення світлового потоку, що падає на розрахункову поверхню, до сумарного потоку від усіх ламп і обчислюється в долях одиниці [8]); 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,  $S = 42 \text{ м}^2$ ;

$h$  – розрахункова висота підвісу,  $h = 2,9 \text{ м}$  (співпадає з висотою стелі, оскільки лампи освітлення закріплюються на стелі);

$A$  – ширина приміщення,  $A = 6 \text{ м}$ ;

$B$  – довжина приміщення,  $B = 7 \text{ м}$ .

Підставимо всі значення у формулу та визначимо індекс 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:  $F = 71689 \text{ Лм}$ .

Для розрахунку будемо використовувати світлодіодні стельові панелі Delux LED Panel 41 44 Вт, світловий потік яких  $F_{л} = 3600 \text{ Лм}$ .

Число ламп визначається за формулою:

$$N = F / F_{л}$$

де:

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		88

F – світловий потік,

F<sub>л</sub> – світловий потік однієї лампи.

Підставимо всі значення у формулу та визначимо індекс приміщення:

$$N = 71689 / 3600 = 19,9 \text{ шт.}$$

Приймаємо необхідну кількість світлодіодних світильників 20 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.

КБПЗ – 2025

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>89</b>

## 9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи підвищення надійності збереження інформації на SSD дисках з використанням технології RAID.

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 підвищення надійності збереження інформації на SSD дисках з використанням технології RAID.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем підвищення надійності збереження інформації на SSD дисках з використанням технології RAID.
- Досліджена система підвищення надійності збереження інформації на SSD дисках з використанням технології RAID.
- На основі отриманих результатів досліджень створена програмна реалізація системи підвищення надійності збереження інформації на SSD дисках з використанням технології RAID.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання підвищення надійності збереження інформації на SSD дисках з використанням технології RAID.

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.

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

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.

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 Visual C#. 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 CRYPTON.

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.

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Хуторний Д.О. Дослідження та програмна реалізація системи підвищення надійності збереження інформації на SSD дисках з використанням технології RAID // Збірник праць молодих науковців ЦНТУ. – Вип. 15. – Кропивницький: ЦНТУ, 2025.

2. Doug Lowe «Networking For Dummies 12th Edition». 2020. – 480 p.

3. Ramon Nastase «Computer Networking: The Beginner’s guide for Mastering Computer Networking, the Internet and the OSI Model». 2018. – 186 p.

4. Russ White & Ethan Banks «Computer Networking Problems and Solutions: An Innovative Approach to Building Resilient, Modern Networks». 2017. – 832 p.

5. Вінтенко Б., Смірнов О., Миронець І., Смірнова Т., Смірнов С. «Імітаційна модель шляхів вхідних даних комп’ютерної інтелектуальної системи підтримки оператора енергоблоку АЕС». *Комбінаторні конфігурації та їхні застосування: Матеріали XXVII Міжнародного науково-практичного семінару, присвяченого 125-річчю Національного університету «Запорізька політехніка» (Запоріжжя-Кропивницький-Київ, 4-6 червня 2025 р.)*. Запоріжжя: НУ «Запорізька політехніка», 2025. С.82-91.

6. Al-Azzeh, J., Ayyoub, B., Mesleh, A., Smirnova, T., Gnatyuk, S., Drieiev, O., Smirnov, O., Dorenskyi, O. «Cloud-Based Information System for Evaluating Caverns in the Process of Blasting Metal Surfaces of Details». *International Review on Modelling and Simulations* 18 (1), 2025. pp. 32-42.

7. Смірнова Т.В., Коноплицька-Слободенюк О.К., Буравченко К.О., Смірнов С.А., Кравчук О.В., Козірова Н.Л., Смірнов О.А. «Дослідження технологій забезпечення кібербезпеки хмарних сервісів IaaS, PaaS та SaaS». *Кібербезпека: освіта, наука, техніка*. 2024. №4(24), С. 6-27.

					ВКРМ-123.25.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

8. 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.

9. 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.

10. 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.

11. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yenchov, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings*, Volume 3530, 2023, pp. 256-265.

12. 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.

13. 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.

14. 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.

15. 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.

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

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

17. 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.

18. 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.

19. 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.

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

21. 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.

22. 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.

23. 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.

24. 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.

25. 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.

26. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», *CEUR Workshop Proceedings* Volume 2608, 2020, Pages 633-645.

27. 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.

28. 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.

29. 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.

30. 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.

31. 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.

32. 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.

33. Smirnov, O., Kuznetsov, A., Kiian, A., Zamula, A., Rudenko, S., Hryhorenko, V., «Variance Analysis of Networks Traffic for Intrusion Detection in Smart Grids», *2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS)*, Kyiv, Ukraine April 17-19, 2019 P. 353-358.

34. 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.

35. 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.

36. 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.

					<b>БКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		96

37. 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

38. 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.

39. 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.

40. 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.

41. 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.

42. 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.

43. 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.

44. 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

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

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).

45. 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

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

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

48. 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.

49. 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.

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

51. 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.

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		98

52. 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.

53. 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.

54. 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.

55. 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.

56. Смірнов О.А., Кавун С.В., Коваленко О.В., Дреєв О.М. Мережні інформаційні технології. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 159 с.

КБГЗ-2025

					<b>ВКРМ-123.25.0067.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		99