

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

« ____ » _____ 2026 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за першим (бакалаврським) рівнем вищої освіти
на тему

**“Програмне забезпечення системи створення та збереження
образів електронних носіїв інформації на основі використання
функцій DeviceIoControl ОС Windows”**

Виконав здобувач вищої освіти

IV курсу, групи КІ-21-1

ОПП «Комп’ютерна інженерія»

спеціальності 123 «Комп’ютерна інженерія»

_____ Р.В. Кадубець

« ____ » _____ 2026 р.

Керівник проекту

кандидат технічних наук, доцент

_____ О.А. Кислун

« ____ » _____ 2026 р.

Рецензент _____

« ____ » _____ 2026 р.

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Освітній ступінь бакалавр
Галузь знань ._12 _____“Інформаційні технології”
Спеціальність 123 “Комп’ютерна інженерія”
Освітньо-професійна (освітньо-наукова) програма “Комп’ютерна інженерія”

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.
_____ Олексій СМІРНОВ
«27» лютого 2026 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ПЕРШИМ (БАКАЛАВРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Кадубцю Руслану Віталійовичу

(прізвище, ім’я, по батькові)

1. Тема роботи *Програмне забезпечення системи створення та збереження образів електронних носіїв інформації на основі використання функцій DeviceIoControl ОС Windows*

2. Керівник роботи *Кислун Олег Андрійович, канд. техн. наук, доцент*

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

затверджені наказом вищого навчального закладу № 133-02 від 27.02.2026 року

3. Строк подання студентом роботи до захисту _____ . .2026 р.

4. Мета та завдання випускної кваліфікаційної роботи: *Метою роботи є розробка програмного забезпечення системи відеоспостереження руху автомобільного транспорту на основі використання елементів бібліотеки DirectShow ОС Windows та API-функцій*

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

1. Призначення та область використання.

2. Перегляд аналогічних існуючих систем.

3. Опис і обґрунтування проектних рішень.

4. Етапи програмування системи.

5. Впровадження системи в промислову експлуатацію.

6. Висновки

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

Структурна схема системи *1 аркуш*

Функціональна схема системи *1 аркуш*

Діаграма процесів *1 аркуш*

Блок-схема алгоритму роботи додатку *4 аркуші*

7. Дата видачі завдання « 27 » лютого 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.03.2026 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2026 р.	
3.	Розробка моделі компонента	20.03.2026 р.	
4.	Розробка структур даних	25.03.2026. р.	
5.	Розробка алгоритмів зв'язку та відображення	30.03.2026. р.	
6.	Програмування алгоритмів	10.04.2026 р.	
7.	Оформлення ПЗ	17.04.2026 р.	
8.	Попередній захист роботи	26.05.2026 р.	

Дата видачі завдання
« 27 » лютого 2026р.

Підпис керівника

_____ Кислун О.А.
(прізвище та ініціали)

Завдання прийнято до виконання
« 27 » лютого 2026 р.

Підпис здобувача

_____ Кадубець Р.В.
(прізвище та ініціали)

АНОТАЦІЯ

Кадубець Р.В. Програмне забезпечення системи створення та збереження образів електронних носіїв інформації на основі використання функцій DeviceIoControl ОС Windows. 123 “Комп’ютерна інженерія”. Центральноукраїнський національний технічний університет. Кропивницький. 2026.

В даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти розроблено програмне забезпечення, яке призначене для автоматизованої системи створення та збереження образів електронних носіїв інформації на основі використання функцій DeviceIoControl ОС Windows.

Метою розробки є програмне забезпечення автоматизованої системи створення та збереження образів електронних носіїв інформації на основі використання функцій DeviceIoControl ОС Windows.

Результат роботи – програмна реалізація автоматизованої системи створення та збереження образів електронних носіїв інформації на основі використання функцій DeviceIoControl ОС Windows.

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

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

Програма може використовуватися на комп’ютер Intel® Celeron/8 Mb/1.2 Gb/SVGA 14” 1Mb або сумісні з ним з ОС Windows.

Програму розроблено в середовищі Delphi.

Ключові слова: образ диска, резервне копіюваннях, цілісність збережених образів, захист збережених образів.

ABSTRACT

Kadubets R.V. Software for creating and saving images of electronic media based on the use of DeviceIoControl OC Windows. 123 the "Computer engineering". Central Ukrainian National Technical University. Kropyvnytskyi. 2026.

In this final qualifying work after the first (by a bachelor) level of higher education software, what creation and maintenance of characters of electronic transmitters of information intended for CAS, is worked out on the basis of the use of functions of DeviceIoControl OC Windows.

The aim of development is CAS of creation and maintenance of characters of electronic transmitters of information software on the basis of the use of functions of DeviceIoControl OC Windows.

A job performance is programmatic realization of CAS of creation and maintenance of characters of electronic transmitters of information on the basis of the use of functions of DeviceIoControl OC Windows.

In the process of prosecution of programmatic model the analysis of existent vehicle and programmatic facilities is executed. All components of the worked out software are described in a complete measure.

Worked out comfortable man-machine interface. The brought instructions over on work with programmatic facilities.

The program can be used on the computer of Intel(Celeron/8 Mb/1.2 Gb/SVGA 14" 1Mb or consonant with him from OS of Windows.

The program is worked out in the environment of Delphi.

Keywords: disk character, reserve printing-downs, integrity of the stored characters, defense of the stored characters.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	2
ВСТУП.....	3
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	7
1.1 Призначення системи.....	7
1.2 Область застосування.....	12
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	14
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим бакалаврським) рівнем вищої освіти.....	15
2.2 Обґрунтування вибору засобів для побудови системи кібербезпеки та мови програмування.....	26
2.3 Розгорнута постановка завдання	31
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	35
3.1 Опис функціонування системи	35
3.2 Розробка структурної схеми.....	46
3.3 Розробка функціональної схеми	52
3.4 Розробка діаграми процесів.....	55
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	57
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	57
4.2 Захист розробленого програмного забезпечення.....	79
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	81
6 ОСНОВНІ ВИСНОВКИ.....	88
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	91

					ВКРБ-123.26.0011.00.00.ПЗ			
Вим.	Арк.	№ докум.	Підп.	Дата	Програмне забезпечення системи створення та збереження образів електронних носіїв інформації на основі використання функцій DeviceIoControl ОС Windows	Літ.	Аркуш	Аркушів
Розроб.	Кадубець РВ.					Б	1	96
Перев.	Кислун О.А..							
Н.контр.	Коваленко А.С.					ЦНТУ КІ-22-1		
Затв.	Смірнов О.А.							

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ВКБР	- випускна кваліфікаційна бакалаврська робота
ЕНІ	- електронні носії інформації
ЖД	- жорсткий диск
ЕНІ	- електронні носії інформації
МНІ	- магнітні носії інформації
ОС	- операційна система
ПЗ	- програмне забезпечення
ПК	- персональний комп'ютер
ТЗ	- технічне завдання
ФС	= файлова система

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

ВСТУП

Актуальність теми. Під час підготовки випускної кваліфікаційної роботи бакалаврського рівня планується створення програмного продукту, що забезпечує функціонування автоматизованої системи формування та збереження образів електронних носіїв інформації. Основою реалізації такої системи є використання можливостей операційної системи Windows, а саме - застосування функції DeviceIoControl, яка дозволяє здійснювати керування та обмін даними з апаратними пристроями.

У сучасному світі інформація набуває статусу одного з найцінніших ресурсів. Розвиток процесів інформатизації суспільства сприяє формуванню високотехнологічного середовища, у якому доступ до достовірних і своєчасних даних часто має більшу цінність, ніж матеріальні ресурси. Швидке отримання необхідної інформації забезпечує суттєву конкурентну перевагу над тими, хто не володіє подібними даними. Водночас потрапляння конфіденційної інформації щодо діяльності фізичної або юридичної особи до сторонніх осіб чи зловмисників може призвести до значних негативних наслідків для її власника.

Після початку повномасштабного військового вторгнення рф в Україну 24 лютого 2022 року держава змушена функціонувати в умовах запровадженого воєнного стану. Поряд із безпосередньою військовою агресією проти України активно ведуться й дії у кіберпросторі, спрямовані на дестабілізацію інформаційної інфраструктури. Згідно зі статистичними даними, інтенсивність кібернетичних атак на державні системи та критично важливі інформаційні ресурси зростає у декілька разів. Зростання складності технологій проведення кібернападів та їх частоти обумовлює необхідність створення і впровадження сучасних механізмів захисту інформації, які забезпечуватимуть її конфіденційність, цілісність і зменшення ризиків виникнення потенційних загроз.

В умовах широкомасштабної збройної агресії, яку рф веде проти нашої

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

країни, постають нові виклики у сфері збереження інформаційних ресурсів. Зокрема, суттєво зростає ризик втрати фізичних робочих документів, архівних матеріалів, а також резервних копій, що зберігаються на магнітних носіях даних. З огляду на це, особливого значення набуває необхідність розробки та впровадження автоматизованих систем, призначених для створення та збереження образів електронних носіїв інформації (цифрових копій дисків). Впровадження таких рішень є важливим для України, оскільки дозволяє захистити дані від фізичного знищення, наслідків кібератак і забезпечити безперервність управлінських та бізнес-процесів в умовах воєнного часу. Крім того, використання подібних систем дає можливість оперативно формувати резервні копії інформації, гарантувати цілісність документів і забезпечувати віддалений доступ до них незалежно від місця перебування користувача.

Образ диска являє собою спеціальний файл, у якому зберігається точна побайтова копія вмісту та структури фізичного носія інформації, такого як CD, DVD, жорсткий диск або USB-накопичувач. Такий файл відтворює не лише самі дані, але й файлову систему, розташування файлів, службову інформацію та завантажувальні записи носія. Найчастіше образ представлений у вигляді одного файлу, наприклад із розширенням .ISO або .IMG. Подібні файли застосовуються при попередній розробці копій даних для їх відновлення у разі втрати, поширенні програмного забезпечення, а також для емуляції роботи фізичного диска в операційній системі. По суті, образ диска є точним цифровим відображенням джерела, що зберігає структуру та розташування інформації у первісному вигляді.

Результативність відновлення інформації під час використання механізмів резервного копіювання ґрунтується на дотриманні низки основних принципів. Передусім технічне обладнання та носії даних повинні перебувати у справному стані, а сама інформація має зберігатися без пошкоджень. Крім того, процес створення резервних копій необхідно здійснювати регулярно відповідно до визначеної періодичності. Важливим є також організоване зберігання копій даних відповідно до обраної користувачем стратегії резервування.

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

Для розміщення резервних копій можуть використовуватися різні типи сховищ, зокрема зовнішні накопичувачі, мережеві системи зберігання даних (NAS), хмарні платформи або спеціалізовані дата-центри. При цьому одним із ключових принципів надійного резервування є зберігання принаймні однієї копії інформації у фізично віддаленому місці, що дозволяє мінімізувати ризик втрати даних при виникненні надзвичайних подій, критичних станів або інцидентів.

Виготовлення електронних образів є важливим етапом інтеграції документів у цифрове середовище та організації ефективного управління інформацією. Збереження даних у вигляді образів електронних носіїв забезпечує їх довготривале зберігання, оперативний доступ і належний рівень захисту. Такий підхід дозволяє формувати структуровані бази даних, підтримувати цілісність документів і забезпечувати швидке відновлення інформації у разі потреби.

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи

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

- обстеження існуючих автоматизованих систем створення та збереження у вигляді образів електронних носіїв інформації;
- дослідження автоматизованих систем створення та збереження у вигляді образів електронних носіїв інформації;
- програмна реалізація автоматизованої системи створення та збереження у вигляді образів електронних носіїв інформації на основі використання функцій DeviceIoControl ОС Windows.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі створення та збереження у вигляді образів електронних носіїв інформації на основі використання функцій DeviceIoControl ОС Windows. Розроблена в процесі виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти (ВКРБ) система є дієздатною та готовою до впровадження на 100%.

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

Зважаючи на вищевикладене, розроблення програмного забезпечення автоматизованої системи створення та збереження у вигляді образів електронних носіїв інформації на основі використання функцій DeviceIoControl ОС Windows є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

К6ПЗ_2026

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

Об'єктом розробки в межах даної випускної кваліфікаційної роботи виступає програмний комплекс для створення інформаційних дублікатів у форматі образів із застосуванням функцій DeviceIoControl ОС Windows. Під терміном образ диска слід розуміти спеціалізований файл, який реплікує повну структуру та масиви даних першоджерела (будь-то розділ HDD, оптичний носій чи дискета). Ключовою характеристикою образу є його інваріантність щодо типу носія: він акумулює всі відомості про розміщення та вміст секторів, що дозволяє ідентично відтворити конфігурацію будь-якого накопичувача, незалежно від специфіки його файлової системи.

1.1 Призначення системи

Функціонування актуальних інформаційних структур критично залежить від стабільності доступу, автентичності та збереження структури даних. Масштабування клієнтської бази, розширення архівів, а також динамічна еволюція технічного й програмного забезпечення актуалізують проблему безперебійної експлуатації ресурсів. З огляду на загрозу незворотної втрати контенту, грамотна побудова процесів реплікації та архівації стає ключовим гарантом безпеки та відкритості інформації / даних.

Для більшості сучасних ІТ-систем ключовими критеріями є постійна наявність, точність та цілісність інформації. Постійне збільшення навантаження на сховища та швидка зміна технологічного стека ускладнюють підтримку сталого доступу. В умовах високих ризиків втрати даних, ефективна стратегія резервування та відновлення стає фундаментальною умовою інформаційної стійкості організації.

Ризики для цілісності даних класифікують за джерелом виникнення на

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

логічні та фізичні. Техногенні катастрофи та вихід із ладу «заліза» генерують близько 45% втрат. Водночас 35% кейсів мають логічну природу: помилкові дії персоналу (понад третина випадків), а також некоректна робота софту (12%) та шкідливе ПЗ (8%). Єдиним дієвим механізмом захисту залишається резервне копіювання — процедура дублювання файлів на альтернативні носії, що гарантує можливість відкату до вихідного стану при пошкодженні оригіналів.

Резервне копіювання охоплює також сегмент мобільних медіа-носіїв, зокрема, зовнішні HDD/SSD та флеш-драйви. На відміну від носіїв малої місткості, вони дозволяють розгортати автономні робочі середовища (Live OS) або використовуватись як фізичні ідентифікатори. Наприклад, певний інформаційний ресурс може залишатися заблокованим до моменту розпізнавання системою конкретного флеш-накопичувача, що виконує функцію цифрового ключа.

У разі виходу з ладу такого пристрою, що трапляється рідше за збої стаціонарних дисків, проте є критичним, власник втрачає доступ до заблокованих ресурсів через відсутність апаратного ідентифікатора. З огляду на доступність сучасних флеш-накопичувачів, доцільно заздалегідь створити дублікат ключа. Найбільш ефективним методом є використання спеціалізованого софту для формування точного образу диска або застосування посекторного копіювання, що дозволяє перенести ідентичну структуру даних на резервний носій.

Недоліки експлуатації CD/DVD-дисків зумовлені такими факторами:

- Чутливість до дефектів: поява мікроподряпин на робочій поверхні часто блокує доступ до інформації.
- Ризик фізичної деформації: через використання некондиційної сировини носій може розлетітися на фрагменти під час обертання на високих швидкостях.
- Обмежений життєвий цикл та низька продуктивність: недовговічність хімічного складу робочого шару та повільне зчитування даних є суттєвими мінусами, які провокують нестабільну роботу ігрового чи мультимедійного софту.

Фізична деградація оптичних носіїв часто спричинена їх інтенсивною експлуатацією. Багато видів спеціалізованого софту та ігрових додатків

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

потребують постійної присутності оригінального диска у зчитувачі для верифікації. Через регулярне механічне навантаження навіть високоякісні екземпляри вкриваються мікротріщинами та стають непридатними. Оптимальним рішенням для запобігання зносу є формування резервної копії або використання програмних емуляторів, які дозволяють монтувати віртуальні образи дисків і працювати з ними, як із фізичними пристроями.

Беручи до уваги зазначені фактори та гетерогенність сучасних засобів зберігання, проєктована система має підтримувати широкий спектр популярних форматів образів. До них належать розширення c2d (WinOnCD), img (Clone CD), bin, nrg (Nero), cif (Easy CD Creator), а також fcd (Virtual CD-ROM) та ще понад тридцять інших специфікацій. Що стосується архітектури процесу, то виділяють два базові підходи до резервування: пооб'єктне (пофайлове) копіювання та повне клонування (дублювання) дискових масивів.

Інструментарій для пооб'єктного резервування та софт для формування образів дисків мають різне функціональне призначення і практично не взаємозамінюються. Оскільки комплексна стратегія захисту передбачає поєднання повного клонування накопичувача з регулярним копіюванням окремих файлів у проміжних циклах, виникає потреба в одночасному використанні декількох програмних продуктів.

Для клонування дискових масивів найчастіше застосовують спеціалізоване ПЗ, що створює точні «знімки» жорсткого диска. Складність архітектури подібних додатків зумовлює те, що їх випуском займаються лідери ІТ-ринку. Оскільки основний функціонал у різних виробників реалізований на належному рівні, основним критерієм при виборі конкретного інструмента стає його продуктивність, а саме — тривалість операції створення бекапу.

Концепція пооб'єктного (пофайлового) копіювання базується на вибіркового резервуванні конкретних даних, визначених користувачем. Попри низьку ресурсомісткість, цей підхід є ефективним переважно для вирішення індивідуальних завдань. З огляду на це, при прокатуванні системи резервного

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

копіювання та відновлення (ВКРБ) зазначену методику не будемо відносити до переліку пріоритетних функціональних вимог.

Принцип селективного (пофайлового) дублювання фокусується на створенні копій лише тих масивів даних, які безпосередньо вказує оператор. Оптимальність використання обчислювальних потужностей при такому підході нівелюється його обмеженістю в масштабах корпоративних рішень, що робить його придатним здебільшого для персонального вжитку. Таким чином, у межах розробки архітектури ВКРБ-системи даний механізм не розглядатиметься, як ключовий елемент функціональної специфікації.

Оволодіння технологіями створення резервних копій є обов'язковим елементом професійної підготовки фахівців ІТ-сфери. Ефективність обраного рішення детермінується низкою параметрів: встановленим рівнем безпеки та обраним інструментарієм (апаратне і програмне забезпечення), кількісними показниками інформаційних масивів (вибір відповідних засобів зберігання) та особливостями архітектури локальної мережі. Організаційна структура мережі, у свою чергу, зумовлює алгоритм систематичного резервування та визначає рівень трудомісткості процесів підтримки цілісності даних.

При значних масивах інформації для реалізації бекапу зазвичай розгортають спеціалізовані програмно-технічні структури, відомі як системи резервного копіювання. Їхній функціонал охоплює не лише автоматизоване дублювання системних ресурсів та клієнтських даних за графіком, а й забезпечення їхньої екстреної регенерації у разі збоїв.

Проте високовартісні програмно-апаратні рішення часто виходять за межі бюджетних можливостей невеликих організацій або приватних осіб. Оскільки обсяг життєво важливої інформації в таких випадках зазвичай помірним, оптимальним з точки зору витрат, є використання побутових HDD-носіїв для формування та підтримки цифрових архівів.

Разом з тим, для представників малого бізнесу чи індивідуальних користувачів експлуатація коштовних програмно-технічних комплексів часто є

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

економічно необґрунтованою. З огляду на відносно невеликі масиви критичних даних, найбільш раціональним рішенням з позиції бюджету стає застосування стандартних HDD-накопичувачів для створення та збереження інформаційних архівів.

Стратегія ізольованого зберігання даних однаково стосується як файлових масивів, так і сформованих образів систем. Мінімальною вимогою є їх розміщення на логічних розділах, відмінних від системного. Для корпоративного сектору регламент безпеки має бути суворішим: принаймні один дублікат ресурсів слід розташовувати на територіально віддаленому майданчику (в іншому офісі чи філії). Такий підхід гарантує збереження корпоративних активів у разі техногенних катастроф, пожеж, стихійних лих або дій зловмисників.

Підсумки аналізу, проведеного під час переддипломної практики, свідчать, що майже 80% організацій, як розробників софту, так і активних користувачів ІТ-рішень, ігнорують регламентоване (згідно з графіком) резервування даних.

Практика архівації даних лише по закінченню звітних періодів чи завершенню робіт не забезпечує належної стійкості інфраструктури. Такий підхід унеможливорює швидке відновлення критичних масивів при виході носіїв із ладу, створюючи передумови для прямих матеріальних збитків. Отже, відсутність системних заходів із захисту власної чи службової інформації створює серйозні ризики для стабільності бізнесу. Процедура створення резервних копій має бути невід'ємною частиною цифрової гігієни, оскільки ціна втрати даних значно перевищує витрати на їх захист.

Отже, створення програмного комплексу для автоматизованого формування образів носіїв є надзвичайно актуальним і затребуваним завданням. Реалізація зазначеного функціоналу в автономному режимі дозволить задовольнити потреби як звичайних користувачів, так і кваліфікованих фахівців у галузі розробки ПЗ, забезпечуючи надійний захист інформаційних активів.

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

1.2 Область застосування

Цільовою аудиторією для впровадження розроблюваної системи є підприємства та установи, діяльність яких пов'язана з розробкою програмного забезпечення або функціонуванням систем електронного документообігу (СЕД) із великими масивами даних. Переваги обраного підходу (використання образів) перед стандартними методами полягають у наступному:

- дистрибуція програмних продуктів: формат образу є оптимальним для розгортання масштабних програмних комплексів або дистрибутивів операційних систем, зокрема через мережеві ресурси;

- віртуалізація: отримані копії можуть виступати в ролі віртуальних накопичувачів для емуляторів та середовищ віртуальних машин;

- клонування конфігурацій: образи дозволяють здійснювати масове розгортання ідентичних робочих місць. Достатньо один раз сконфігурувати еталонну систему з усіма драйверами та софтом, щоб потім тиражувати цей «знімок» на парк аналогічного обладнання;

- гарантія цілісності при бекапі: традиційні утиліти часто не мають доступу до системних файлів або заблокованих структур. Образ диска, натомість, зберігає повну побітну структуру носія;

- транспортування ПЗ: це зручний спосіб передачі великих пакетів даних та інсталяційних образів ОС через інтернет-канали;

- інтеграція з віртуальними середовищами: образи дисків інтегруються в системи віртуалізації як повноцінні пристрої збереження; ;

- масштабування систем: метод ідеально підходить для уніфікації ПЗ на корпоративних комп'ютерах, дозволяючи переносити налаштоване середовище з одного ПК на багато інших без повторної інсталяції.

Завдяки модульній структурі, програмне рішення можна легко налаштувати під вимоги замовника без залучення значних ресурсів на доопрацювання. Це відкриває можливості для:

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

- врахування унікальних бізнес-процесів і структури обміну документами на підприємстві;
- розгортання системи у гнучких форматах: від окремих робочих місць до масштабних серверних платформ.

К6ПЗ_2026

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

Об'єктом розробки в межах даної кваліфікаційної роботи є спеціалізований програмний комплекс для автоматизації процесів «імеджингу» (створення образів дисків) із залученням функцій DeviceIoControl ОС Windows. Кінцевий продукт має забезпечувати реалізацію такого функціоналу:

- регламентоване дублювання: автоматичне формування образів згідно з попередньо налаштованим графіком або за ініціативою оператора;
- мультиплатформенне збереження: експорт резервних копій на різноманітні типи носіїв через інтеграцію із засобами запису оптичних дисків (CD/DVD), магнітними стрічками (стрімерами) та іншими пристроями;
- аудит процесів: ведення детальних лог-файлів (журналювання) усіх операцій з архівації та подальшої реанімації даних;
- інтелектуальний моніторинг: аналіз динаміки змін у цільових файлових структурах і запуск процедури копіювання при досягненні встановленого порогу модифікацій.

Серед додаткових можливостей ПЗ варто виділити:

- фільтрацію файлів для бекапу на основі заданих масок або типів розширень;
- багатопотокове копіювання з одночасним виводом на кілька різних накопичувачів;
- керування пріоритетністю завдань через механізм черги.

Таким чином, дослідження спрямоване на підвищення стійкості інформаційної інфраструктури суб'єктів господарювання, адже оперативна реанімація системи після програмних чи апаратних помилок напряму залежить від наявності вчасно створених архівів.

Проектування архітектури системи, визначення її структурних компонентів, а також формування технічних вимог і функціональних обмежень

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

потребує попереднього ґрунтовного вивчення наявних програмних аналогів.

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

В процесі проходження переддипломної практики було проведено моніторинг та систематизацію найбільш затребуваних програмних рішень, що використовують функції DeviceIoControl ОС Windows та API-функцій для генерації дискових образів. Нижче подано результати порівняльної оцінки виявлених аналогів.

Варто зауважити, що інструментарій для пооб'єктної архівації та софт для «імеджингу» (створення знімків дисків) мають різну архітектурну логіку та функціональне призначення. Оскільки раціональна стратегія захисту інформації передбачає комбінування повного дублювання носія із селективним копіюванням файлів між циклами, на практиці виникає потреба в експлуатації комплексу різних програмних засобів.

У нашому регіоні домінують спеціалізовані інструменти «імеджингу», створення яких під силу лише потужним розробникам програмного забезпечення. Попри ідентичність основного призначення, ефективність таких утиліт доцільно оцінювати за швидкістю обробки даних. Критеріями диференціації при підборі ПЗ виступають економічна складова та розширений інструментарій: можливість запуску з Preinstallation Environment (PE) або Live CD, сегментація архівних файлів, доступ до окремих елементів всередині образу без його повної регенерації, а також зручність користувацького середовища.

Наразі сегмент ПЗ для «імеджингу» дисків переживає трансформацію: на зміну застарілим інструментам минулих десятиліть прийшли потужні мультифункціональні платформи. Гідну конкуренцію традиційним брендам складають такі рішення, як Paragon Hard Disk Manager та Clonezilla (для

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

прихильників відкритого ПЗ). Особливе місце посідає Acronis, що інтегрує функції бекапу з кіберзахистом. Порівняно з попередниками, ці додатки значно швидше опрацьовують великі масиви даних та надають розширені можливості відновлення на «голе залізо» (Dissimilar Hardware Restoration).

Acronis Cyber Protect Home Office: Комплексне рішення для захисту та резервного копіювання.

Цей програмний комплекс являє собою інтегровану платформу, яка поєднує традиційне створення образів дисків із сучасними механізмами кібербезпеки. Основна перевага системи – застосування розумних алгоритмів, що забезпечують захист архівованих даних у реальному часі та контроль цілісності інформації.

Переваги:

- завдяки модулю Universal Restore користувач може відновлювати операційні системи на апаратних конфігураціях, які відрізняються від оригінальних, без додаткової підготовки;

- опція «гарячого» клонування дозволяє створювати точні копії системного розділу без призупинення роботи користувача та активних програм.

Недоліки:

- висока складність архітектури призводить до значного використання ресурсів комп'ютера, включно з оперативною пам'яттю та центральним процесором;

- передплатна модель ліцензування може бути фінансово обтяжливою для малих підприємств та індивідуальних користувачів.

Інтеграція та функції DeviceIoControl OC Windows та API: продукт тісно взаємодіє з Windows Volume Shadow Copy Service (VSS). Це забезпечує створення надійних знімків дисків навіть під час активного використання файлових систем та баз даних.

Висновок: Acronis Cyber Protect Home Office є оригінальним розв'язанням проблеми для користувачів, які потребують одночасно надійного резервного

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

копіювання та превентивного захисту від кіберзагроз з можливістю гнучкого відновлення на різних апаратних конфігураціях.

Macrium Reflect: ПЗ для резервного копіювання дисків.

Macrium Reflect – це багатофункціональний інструмент для створення образів дисків і резервного копіювання, який дозволяє користувачам надійно відновлювати дані та системи у разі збою. Продукт поєднує простий інтерфейс із потужними механізмами управління образами та інтеграцією з операційною системою.

Переваги:

- підтримка дійсного клонування системних розділів та створення образів дисків із мінімальними перериваннями роботи користувача;
- наявність функцій швидкого відновлення системи і можливість завантаження з образу за допомогою Rescue Media;
- підтримка сучасних файлових систем Windows (NTFS, ReFS) та інтеграція з Windows VSS, що дозволяє створювати надійні знімки дисків під час роботи системи.

Недоліки:

- професійні функції резервного копіювання та клонування доступні лише у платній версії;
- для великих корпоративних середовищ може знадобитися значна обчислювальна потужність та додаткові ресурси AOMEI Backupper (користувацьке рішення).

Інтеграція та функції DeviceIoControl ОС Windows та API: продукт використовує Windows Volume Shadow Copy Service (VSS) для забезпечення консистентності даних у резервних копіях та підтримує інтеграцію з сучасними дисковими драйверами і системними службами Windows.

Висновок: Macrium Reflect – ефективний інструмент для персональних користувачів та малих організацій, які прагнуть надійного резервного копіювання, швидкого відновлення системи та збереження даних у безпечному вигляді.

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

Clonezilla (безкоштовне рішення для клонування та резервного копіювання дисків).

Clonezilla – це відкритий програмний комплекс для створення образів дисків і клонування розділів, орієнтований на ефективне резервне копіювання та масове розгортання операційних систем. Основна перевага – низьке споживання ресурсів та здатність працювати з різними типами файлових систем.

Переваги:

- підтримка масового клонування дисків і розділів, що дозволяє швидко відновлювати робочі середовища на багатьох комп'ютерах одночасно;

- працює з великою кількістю файлових систем: ext2/3/4, NTFS, FAT, exFAT, HFS+, а також з мережевими файловими системами через SSH, Samba, NFS;

- не вимагає встановлення та може запускатися з Live-CD/USB, що дозволяє працювати незалежно від операційної системи на комп'ютері.

Недоліки:

- відсутність графічного інтерфейсу користувача в стандартному наборі; робота переважно через текстовий інтерфейс (CLI), що може бути складним для новачків;

- обмежена інтеграція з сучасними системними API Windows; для роботи з VSS або динамічними томами потрібно додаткове налаштування.

Інтеграція та функції DeviceIoControl OC Windows та API: Clonezilla використовує базові системні утиліти Linux для доступу до дисків і розділів, а також підтримує мережеві протоколи для зберігання образів. Завдяки цьому забезпечується універсальність та незалежність від платформи.

Висновок: Clonezilla – оптимальний вибір для досвідчених користувачів та IT-адміністраторів, які потребують безкоштовного і надійного інструменту для клонування та резервного копіювання, особливо у середовищах із великою кількістю комп'ютерів або різноманітними файловими системами.

R – Drive Image є ефективною утилітою для створення образу диска і

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

резервного копіювання даних. Файл образу диска, що містить точну, «байт у байт» копію жорсткого диска, розділа або логічного диска, може бути створений без перезавантаження комп'ютера та з різним рівнем стискування даних. Створений образ диска може бути збережений на носіях різного виду, включаючи такі знімні носії, як CD – R (W)/DVD, Iomega Zip і Jazz-диски.

Можливості R – Drive Image:

- Простий і зручний інтерфейс користувача, який не потребує поглиблених знань з управління комп'ютером.

- Виконання операцій «на льоту»: образи диска створюються без необхідності зупинки і перезавантаження операційної системи Windows. В процесі створення образу і до його завершення усі інші операції запису зберігаються в кеш-пам'яті. Відновлення образу також здійснюється «на льоту», за винятком системного розділу. При відновленні системного розділу необхідно перезавантажити програму в псевдографічний режим безпосередньо з Windows або за допомогою заздалегідь створених завантажувальних дисків будь-якого типу.

- Стискування файлів образів. Для економії дискового простору файли образів можуть бути стиснені.

- Підтримка переносних медіа-пристроїв. Файли образів можуть бути збережені на переносні медіа-облаштування типу CD – R (W)/DVD, Iomega Zip і Jazz-диски.

- Підтримка мереж в завантажувальній версії. Завантажувальна R – Drive Image версія підтримує збереження образу диска на віддаленому комп'ютері і його відновлення з віддаленого комп'ютера на локальний в мережах Microsoft (CIFS-протокол).

- Розширений список підтримуваних пристроїв в завантажувальній версії. Додані нові апаратні компоненти комп'ютера, підтримувані в завантажувальній версії R, Drive Image.

- Образ диска може бути приєднаний як віртуальний диск, доступний тільки для читання. Уміст такого диска може бути переглянутий і скопійований.

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

- Відновлення окремих файлів і тек. З образу можуть бути відновлені окремі файли і теки або при виконанні відповідної дії, або при під'єднанні образу, як віртуального диска.

- Розподіл файла образу. Файл образу диска може бути розділений на декілька файлів при нестачі місця для його зберігання на одному носії.

- Захист образу диска. Файл образу може бути захищений паролем і містити додаткові коментарі.

- Створення нових розділів. Дані образу диска можуть бути відновлені на вільний дисковий простір. Розмір відновлювального розділу може бути змінений.

- Зміна розділів. Дані образу диска можуть бути відновлені в будь-який існуючий розділ. Програма може видати існуючий розділ і відновити образ диска на звільнений вільний простір.

- Копіювання диска на диск. Цілий диск може бути скопійований на будь-який інший, такий же або більшої місткості.

- Перевірка створеного образу. Файл образу диска може бути перевірений на предмет коректності його створення до того, як він збережений, і/або відновитися.

- Розклад. Задача створення образу диска може бути встановлена і виконана в певний час в автоматичному режимі.

- Створення скрипта для частих і автоматичних дій. Інтерфейс програми дозволяє створювати скрипти для автоматичного створення файлу образу і додавання даних в існуючий файл образу диска. Скрипти виконуються з командного рядка і такий рядок може бути включений у будь-який командний файл.

- Повідомлення про виконані операції. Програма автоматично створює і відправляє повідомлення по електронній пошті, а також може запустити зовнішню утиліту, коли образ диска успішно створений або операція по його створенню не виконана.

- Підтримка файлової системи ReFS (Стійка Файлова Система), нової

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

локальної файлової системи, яку Мікрософт ввів в ОС Windows 2012 Server. Підтримуються усі дискові дії, за виключенням змінного розміру розділу.

- Повна підтримка розмітки диска стандарту GPT. R – Drive Image може створювати GPT диски, змінювати їх розміри і міняти розмітку диска під час операцій копіювання/відновлення.

UltraISO Premium Edition – приклад однієї з кращих програм, призначених для повноцінної роботи з образами дисків. Їй під силу практично усе: не розпаковуючи образи, надає можливість редагувати дані, а також видаляти їх частину, є функція створення образів і запису їх на носії.

Програма працює як із загальноприйнятим стандартом образів ISO, так і з форматами образів, створеними в інших популярних програмах: BIN/CUE, IMG/CCD/SUB, MDF/MDS, C2D, NCD, VDI, CIF, NRG, BWI/BWT, LCD, CDI, PDI, GCD/GI, VC4/000, GI, TAO/DAO, CIF, VCD, VaporCD, тощо. Окрім цього, користувач знайде багато інших корисних функцій для роботи з образами дисків.

Попри те, що UltraISO Premium Edition має свою функцію створення віртуального приводу, вона також інтегрується з іншими популярними програмами з аналогічними можливостями, наприклад, Alcohol 120 % і DaemonTools. Інтерфейс програми має безліч локалізацій, у тому числі і українську.

Можливості програми:

- Пряме редагування утримуваного файлу образа.
- Безпосереднє витягання файлів або тек з файлу образа.
- Можливість додавати, видаляти, створювати нові теки, перейменовувати файли утримуваного файлу образа.
- Створення ISO-образа з будь-якого документу з жорсткого диска.
- Створення CD/DVD образа, з підтримкою автозавантаження.
- Підтримка більшості відомих і невідомих форматів (.ISO, .BIN, .IMG, .CIF, NRG, .BWI і інші), а також їх конвертація у формат ISO.
- Підтримка ISO 9660 рівня 1/2/3.
- Автоматична оптимізація структури файла образа.

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

- Подвійне вікно призначеного для користувача інтерфейса.

Створення резервної копії жорсткого диска ноутбука за допомогою програми **Acronis True Image 2013**. Це інтегрований пакет ПЗ для резервного копіювання і відновлення даних, що забезпечує захист системи, файлів і тек. Вона дозволяє клонувати диски, створювати завантажувальні носії (CD/USB) для відновлення ОС і працює з більшістю обладнань зберігання, включаючи локальні диски і мережеві сховища. Ключові можливості Acronis True Image 2013:

- Резервне копіювання: створення повних образів системи або окремих файлів/тек з можливістю автоматичного розкладу.
- Відновлення даних: швидке відновлення працездатності Windows з резервної копії, включаючи роботу з «голого заліза» (bare - metal).
- Клонування дисків: перенесення усієї інформації, налаштувань і ОС з одного жорсткого диска на інший.
- Завантажувальний носій: створення дисків аварійного відновлення для доступу до даних, коли ОС не завантажується.
- Зручний інтерфейс: майстер-інтерфейс в стилі ОС Windows, що полегшує налаштування, особливо у версії Lite.

Програма убезпечує дані, захищаючи їх від втрати при поломці жорсткого диска або випадковому видаленні.

Active@ Disk Image – програма, призначена для створення і копіювання образів жорстких дисків. Це використовується при копіюванні даних з комп'ютера на комп'ютер, а також при резервному копіюванні даних. Програма працює в чистому DOSi, тобто, із завантажувальної дискети.

Образ резервування може бути збережений на USB-накопичувачі, в мережному сховищі, записаний безпосередньо на CD/DVD/BR-диски або збережено на інших медіа-носіях.

Ця програма робить точну копію будь-яких дисків (HDD, USB, КОМПАКТ-ДИСК, DVD, Blu-промінь, тощо), ПК і зберігає образ в теці.

Disk-зображення можна використати для відновлення чи модернізації цілих

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

disk-дублювань. У разі комп'ютерної відмови, резервне зображення можна використати, щоб повернути обчислювальну систему або знайти та відновити будь-які необхідні файли зсередини зображення.

Програма створює два типи дискових образів:

- резервний образ диска використовується для резервування цілого комп'ютера або будь-якого комп'ютерного диска і містить тільки використаний простір диска;

- сирий образ диска містить ідентичну копію кожного сектора диска. Цей образ може бути використаний в якості резервування не Windows систем для подальшого відновлення даних.

Можливості програми:

- Створення і відновлення образу диска без перезавантаження комп'ютера.
- Вибираються тільки ті сектори, які містять використовувану інформацію.
- Збереження образу диска на локальний диск, мережевий диск або прямий запис на чистий диск.

- Створення точного посекторного образу для резервування розділів невідомих файлових систем.

- Створення образу диска будь-яких жорстких дисків (IDE, SCSI, USB, FireWire), SSD або флопі-диска.

- Створення ISO-образу з оптичних дисків з прямим записом на інші диски.

- Резервування даних з базових дисків або динамічних дисків.

- Налаштування стискування образу для балансу між розміром архіву і швидкістю запису.

- Створення образу декількох дискових розділів одночасно і їх зберігання в єдиному архіві.

- Відновлення вибраних дисків, розділів або індивідуальних файлів з архіву.

- Запуск Active@ Disk Image з Active@ Boot Disk для виконання усіх операцій з образами незалежно від ОС.

- Автоматичне створення образу дисків з використанням планувальника

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

задач.

- Прості операції з використанням Майстрів.
- Розподіл образу диска на декілька файлів.
- Використання функції захисту створюваних образів паролем.
- Легке переміщення системних і програмних файлів на декілька робочих станцій.

Macrium Reflect – програма розроблена для створення і відновлення образів жорсткого диска і резервування даних. Програма створює образ ОС Windows під час її використання і не вимагає для цього перезавантаження комп'ютера або якихось ще додаткових дій. Крім того, можливий перегляд образу диска і планування бекапів.

Програма має високу швидкість створення образу диска. Образ розміром в 12 Gb даних буде змонтований всього за 16 хвилин. При цьому, розмір стисненого бекапа буде всього 8 Gb. Надмірні і повні образи дисків можуть бути відновлені і змонтовані в якості віртуальних дисків в Провіднику Windows.

Можливості програми:

- Створення образу як усього диска, так і окремих тек і файлів.
- Вибір місця зберігання готового образу (на локальному жорсткому диску, на мережевому диску або оптичному носії).
- Створення образу системи «на льоту» без перезавантаження ОС Windows.
- Вбудований планувальник резервування даних.
- Створення завантажувального диска, використовуючи Linux або BartPE.
- Верифікація готових образів.
- Використання сервісу копіювання Microsoft Volume Shadow (VSS).
- Висока швидкість роботи і міра стискування.
- Забезпечення безпеки з використанням AES 256-бітового шифрування.
- Можливість пакетного запису на CD/DVD.

Отже, програм пофайлового копіювання критеріїв для вибору значно більше, більше і самих програм. Розробка подібного програмного забезпечення

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

доступна програмістові середньої руки, тому багато хто створює програми "під себе", згодом виставляючи свої продукти як freeware (безкоштовне програмне забезпечення).

Великий перелік програм, з одного боку, ускладнює пошук (потрібно перебрати декілька десятків однотипних продуктів), з іншого – можливість знайти практично ідеальний для необхідного виду діяльності інструмент резервного копіювання. Сучасна програма повинна обов'язково підтримувати наступні можливості:

- копіювання даних з довільним ступенем стиснення в архів загальновідомого формату (звичайно це ZIP);
- проведення копіювання на декілька різних носіїв (жорсткий диск, віддалений FTP-сервер, CD-R (RW), DVD-диски й інші носії), по можливості, одночасно;
- архівування чи просто копіювання даних автоматично за розкладом, який встановлює користувач;
- підтримування роботи з локальною мережею;
- здатності захисту архівів паролем;
- підтримка різноманітних операційних систем.

Саме ці критерії необхідно запозичити для розробки нашої системи.

Аналіз існуючого ринку програмних засобів для забезпечення цілісності інформації дозволило визначити концептуальний орієнтир для майбутньої розробки. Найбільш відповідним прототипом було обрано архітектурні рішення, реалізовані у функціоналі Norton Backup (File History). На основі аналізу цієї системи було виокремлено ключові характеристики, що стануть фундаментом нашого проекту:

- мінімальна трудомісткість процесів розгортання та експлуатації;
- висока адаптивність конфігураційних параметрів під потреби користувача;
- оперативність виконання критичних операцій з даними.

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

Аналіз ринку показав, що суттєвим мінусом професійних систем резервування є значні ліцензійні витрати. Саме цей економічний бар'єр визначає актуальність розробки альтернативного ПЗ, орієнтованого на бюджетний сегмент користувачів. Аналіз також виявив такі системні недоліки:

- комерційне ПЗ зазвичай постачається як монолітний «комбайн», що зумовлює його високу ціну. Проте для багатьох користувачів оптимальним рішенням було б використання окремих спеціалізованих компонентів, що зараз технічно та фінансово недоступно;

- відсутність гнучких інтерфейсів для взаємодії з уже розгорнутими на об'єктах системами іншого призначення. Будь-яка адаптація потребує додаткових інвестицій у супровід від розробника.

Відтак, розробка власної системи, що базується на модульному принципі та відкритій архітектурі, є логічним кроком для усунення цих бар'єрів при збереженні високої ефективності захисту даних.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Процес проектування програмного забезпечення орієнтований на роботу в середовищі сучасних 64-бітних версій ОС Windows (10, 11 та Windows Server) із залученням механізмів об'єктно-орієнтованого програмування та компонентів ActiveX. Такий підхід забезпечує високий рівень інтерактивності та візуальної зрозумілості інтерфейсу, що відповідає користувацькому досвіду в даній операційній системі.

Для реалізації критичного функціоналу задіяно низькорівневий інструментарій Windows API. Даний набір базових інтерфейсів є найбільш прямим методом взаємодії прикладного софту з ядром ОС, забезпечуючи максимальну продуктивність і контроль над системними ресурсами, поступаючись у глибині доступу лише драйверам пристроїв. Ми використовуємо Win64.

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

Розробка базується на використанні 64-розрядного інтерфейсу Windows API, що є стандартом для актуальних версій операційної системи. Основна логіка взаємодії з ядром та системними службами реалізована через звернення до фундаментальних бібліотек, таких як kernel32.dll (у 64-бітних системах виконує функції базового рівня), advapi32.dll та user32.dll - для керування графічним інтерфейсом. Застосування системних викликів дозволяє оптимізувати продуктивність ПЗ та суттєво скоротити цикл розробки.

Як інструментальне середовище обрано Embarcadero Delphi, що використовує однойменну мову програмування. Даний вибір аргументований високою швидкістю компіляції та потужними можливостями візуального проектування. Мова Delphi поєднує в собі строгу типізацію та ефективність коду, що дозволяє створювати надійні прикладні рішення у стислі терміни без втрати якості виконання.

Застосування ООП-інструментарію мови Delphi дозволяє структурувати складні об'єкти проектування, перетворюючи ресурси ПК та мережеві вузли на логічні класи. Використання модифікаторів доступу забезпечує захищеність внутрішньої логіки об'єктів, що є критично важливим для моделювання систем керування реального часу, де інформаційний обмін між модулями має бути чітко регламентованим.

Фундаментальною перевагою середовища є баланс між рівнями розробки. У той час як більшість сучасних мов програмування мають виражений крен у бік високорівневих інтерфейсів (що ускладнює роботу з «залізом») або низькорівневого коду (що уповільнює створення GUI), Delphi дозволяє однаково ефективно працювати в обох напрямках, інтегруючи мережеві протоколи безпосередньо у візуальні компоненти.

Таким чином, вибір мови Delphi диктується можливістю ефективного опису архітектури об'єкта та його зовнішніх зв'язків, що суттєво прискорює цикл розробки якісного ПЗ. Технічна реалізація процедур зчитування та запису інформації базуватиметься на використанні системного виклику DeviceIoControl.

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

Даний інтерфейс забезпечує пряму комунікацію з драйверами через керуючі коди (IOCTL), що є більш продуктивним методом порівняно зі стандартними файловими функціями API. Використання IOCTL-кодів дозволяє програмі безпосередньо взаємодіяти з апаратною частиною: наприклад, отримувати геометрію диска, статус носія або ініціювати сервісні команди, що є критично важливим для коректного створення дискових образів.

Використання DeviceIoControl відкриває доступ до розширеного функціоналу керування накопичувачами, включаючи отримання детальної специфікації пристроїв, низькорівневе форматування, адміністрування енергоспоживання та конфігурування портів. Ці специфічні операції неможливо реалізувати через стандартні уніфіковані виклики ReadFile або WriteFile.

Архітектура програмного забезпечення базується на інтеграції стандартних бібліотек Delphi, що виконують такі завдання:

- Vcl.Menus: компонентна база, що забезпечує функціонування ієрархічних та контекстних меню, а також інтерактивних елементів панелі стану.
- Vcl.Dialogs та Winapi.Messages: модулі для генерації системних повідомлень та керування модальними діалоговими вікнами.

Крім того, застосування стандартної бібліотеки WinSock гарантує кросплатформову сумісність мережевих інтерфейсів. Оскільки протоколи та сигнатури функцій WinSock ідентичні для більшості мов програмування (зокрема C/C++), це суттєво спрощує портування коду та забезпечує високу схожість мережевої логіки в середовищах Windows та Linux.

Функціональна цілісність системи забезпечується інтеграцією стандартних бібліотек RTL та VCL, що дозволяє мінімізувати обсяг низькорівневого коду та гарантує стабільність роботи:

- System.SysUtils: центральний модуль, що містить інструментарій для математичних обчислень, перетворення типів даних (число-рядок), обробки форматів дати/часу та реалізації специфічних гіперболічних функцій.
- System.Classes: базовий модуль для архітектурної побудови класів, що

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

забезпечує механізми потокового введення-виведення та управління колекціями об'єктів.

- System.Variants: бібліотека для безпечного маніпулювання динамічними типами даних (Variant) та їхнього приведення до статичних типів.

- Vcl.Graphics: спеціалізований інтерфейс для рендерингу графічних об'єктів, управління канвою (Canvas) та растровими ресурсами.

- Vcl.Forms: ключовий модуль управління віконним інтерфейсом. Він забезпечує життєвий цикл екранних форм, обробку подій введення з клавіатури, виведення системних звітів про стан процесів та організацію діалогового середовища з оператором.

Методологія створення системи передбачає послідовне виконання етапів проектування відповідно до нормативних вимог:

- виокремлення базових функціональних можливостей продукту;
- ідентифікація конкретних задач у межах кожної функції;
- формування деталізованого переліку підзадач.

Програмна реалізація здійснюється через модульну організацію, де головний керуючий сегмент взаємодіє з бібліотеками підпрограм. Функціонал загального призначення охоплює типові обчислювальні процеси: пошук інформаційних активів, контроль доступу, перетворення типів даних, а також формування пакетів технічних звітів та документального супроводу процесів резервування.

При проектуванні системи було дотримано базових принципів розробки:

- композиції: формування складної структури шляхом інтеграції автономних підсистем;
- єдності: спільна цілеспрямованість усіх елементів архітектури;
- відкритості: здатність до взаємодії із зовнішніми програмними продуктами та легкість додавання нових модулів;
- універсальності: придатність системи для вирішення широкого спектра прикладних завдань;

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

- інформаційної єдності: використання уніфікованих протоколів та структур даних у межах всього комплексу;

- інваріантності: стійкість алгоритмів до варіацій вхідних параметрів та технічних умов роботи.

Розкриємо зміст кожного з наведених принципів докладніше.

- Принцип композиції (системної декомпозиції). Цей постулат передбачає формування цілісного програмного середовища шляхом об'єднання автономних, функціонально завершених модулів. У контексті даної роботи це означає можливість незалежної розробки та тестування окремих підсистем (наприклад, модуля взаємодії з драйвером та інтерфейсного блоку) з їх подальшою безшовною інтеграцією в єдину структуру. Такий підхід значно спрощує процес налагодження та дозволяє масштабувати систему без докорінної переробки її ядра.

- Принцип єдності (цільової когерентності). Всі структурні елементи архітектури підпорядковані єдиній глобальній меті - забезпеченню надійного резервування та оперативного відновлення даних. Кожна процедура, від низькорівневого виклику DeviceIoControl до формування звіту, працює як частина злагодженого механізму. Це виключає виникнення конфліктних операцій та дублювання функцій, забезпечуючи високу синергію між програмними компонентами.

- Принцип відкритості (екстенсивності). Архітектура системи проектується з урахуванням можливості її майбутньої еволюції. Принцип відкритості гарантує, що нові функціональні модулі можуть бути додані з мінімальними витратами ресурсів. Крім того, наявність стандартизованих інтерфейсів взаємодії дозволяє системі ефективно комунікувати із зовнішніми програмними продуктами, сторонніми утилітами або автоматизованими системами управління вищого рівня.

- Принцип універсальності (функціональної гнучкості). Система розробляється як багатоцільовий інструментарій, здатний забезпечити здійснення великого кола задач захисту інформації. Завдяки високому рівню абстракції, програмний комплекс може бути адаптований для роботи з різними типами носіїв,

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

файловими системами та сценаріями експлуатації - від індивідуального бекапу на персональному комп'ютері до обслуговування масивів даних у корпоративних мережах.

- Принцип інформаційної єдності (стандартизації даних) У межах всього комплексу впроваджується єдиний стандарт обробки, передачі та представлення даних. Використання уніфікованих протоколів обміну та структурованих форматів даних дозволяє уникнути помилок інтерпретації інформації при її переході між різними модулями системи. Це критично важливо для збереження цілісності образу диска на всіх етапах його життєвого циклу.

- Принцип інваріантності (алгоритмічної стійкості). Ключові алгоритми системи зберігають свою коректність та ефективність незалежно від варіативності вхідних умов. Це означає, що механізми копіювання залишаються стабільними при зміні обсягів даних, технічних параметрів обладнання або параметрів файлової системи. Інваріантність забезпечує надійність роботи ПЗ у динамічних технічних середовищах, гарантуючи передбачуваний результат за будь-яких обставин.

2.3 Розгорнута постановка завдання

У процесі виконання ВКРБ передбачається реалізація комплексу взаємопов'язаних завдань, спрямованих на розроблення та обґрунтування програмної системи відповідно до обраної тематики дослідження.

Першочерговим етапом є визначення сутності та змісту розробки, що відповідає темі ВКРБ, отже потрібно детально проаналізувати предметну область, сформулювати основну ідею майбутнього програмного продукту та аргументовано обґрунтувати актуальність обраної теми. Актуальність визначається сучасними потребами розвитку інформаційних технологій, необхідністю автоматизації певних процесів.

Наступним важливим кроком є визначення функціонального призначення програмної системи та встановлення меж її використання. У межах цього етапу

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

формулюються основні завдання, які повинна виконувати система, визначається коло потенційних користувачів, а також окреслюється сфера її практичного застосування. Це дозволяє більш чітко визначити вимоги до майбутнього програмного продукту та сформувати уявлення про його функціональні можливості.

Важливою складовою підготовчого етапу є аналіз існуючих програмних продуктів, алгоритмів та систем-аналогів, які використовуються для розв'язання подібних задач. У процесі такого аналізу необхідно дослідити їх функціональні можливості, принципи роботи, переваги та недоліки. Це дозволить виявити переваги існуючих рішень, а також виявити їх обмеження. Врахування цих факторів у подальшому сприятиме створенню більш ефективної, зручної та функціональної системи.

Спираючись на одержанні аналітичні дані, створюється технічне завдання на розроблення програмного забезпечення системи. У технічному завданні визначаються основні вимоги до функціональності, продуктивності, надійності та зручності використання програмного продукту. Також встановлюються вимоги до структури системи, її модулів, інтерфейсу користувача та умов експлуатації.

Завершальним етапом підготовчої частини є вибір програмних засобів, інструментальних платформ та мов програмування, які будуть використані для реалізації розроблюваної системи. Вибір таких засобів повинен бути обґрунтований з урахуванням їх функціональних можливостей, ефективності, поширеності, сумісності з іншими технологіями та доцільності застосування для реалізації поставлених задач. Раціонально обрані інструменти розробки сприятимуть створенню якісного програмного продукту, який відповідатиме сучасним вимогам до програмних систем.

У межах виконання ВКРБ необхідно розробити та обґрунтувати методологію формування архітектури програмної системи. При цьому слід визначити основні принципи, які будуть покладені в основу її проектування та реалізації. Особливу увагу потрібно приділити встановленню обмежень, що

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

можуть впливати на процес розроблення програмного забезпечення, зокрема технічних, програмних, ресурсних або організаційних. Обґрунтування таких обмежень дає змогу забезпечити коректність проектування та стабільність подальшого функціонування системи.

Наступним етапом є детальний опис принципів функціонування створюваної системи. Для цього необхідно розробити функціональну та структурну схеми, які відображатимуть загальну організацію програмного продукту, його основні модулі та взаємозв'язки між ними. У процесі розроблення цих схем потрібно визначити їх складові елементи, описати їх призначення та аргументувати необхідність застосування в межах створюваної системи. Окрім цього, необхідно сформулювати діаграму взаємодії процесів, яка наочно відображатиме обмін даними між окремими компонентами системи та логіку їх спільної роботи.

Важливим етапом виконання роботи є безпосередня розробка ПЗ системи, яке повинне забезпечувати реалізацію поставлених у технічному завданні функцій. У процесі розробки потрібно підготувати алгоритми роботи основних програмних модулів, а також побудувати відповідні блок-схеми, що відображатимуть послідовність виконання операцій. Для забезпечення кращого розуміння принципів роботи системи слід виконати покроковий опис алгоритмів з поясненням логіки їх функціонування та ілюстрацією фрагментами програмного коду.

Окрему увагу необхідно приділити забезпеченню інформаційної безпеки створюваного програмного продукту. З цією метою слід визначити оптимальний алгоритм захисту програмного забезпечення від несанкціонованого доступу. Обраний метод захисту має бути обґрунтований з урахуванням вимог безпеки, надійності та ефективності.

Завершальним етапом роботи є опрацювання рекомендацій щодо впровадження створеної системи у промислову експлуатацію. Такі рекомендації повинні містити опис основних етапів інтеграції програмного продукту в існуюче

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

програмно-технічне середовище, умови його ефективного використання.

Задля підвищення наочності та кращого сприйняття результатів виконаної роботи необхідно підготувати графічні матеріали, які відображатимуть основні аспекти розробки. До пояснювальної записки додаються ілюстративні матеріали у вигляді семи аркушів формату А4, що містять схеми, діаграми, блок-схеми алгоритмів та інші графічні елементи, які демонструють структуру, принципи функціонування та результати розроблення програмної системи.

К6ПЗ_2026

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

Відповідно до технічного завдання у процесі виконання випускної кваліфікаційної роботи передбачено розроблення програмного забезпечення автоматизованої системи збереження інформації у вигляді образів носіїв даних. Реалізація системи здійснюватиметься з використанням функцій DeviceIoControl ОС Windows, що забезпечують взаємодію програмного забезпечення з ресурсами операційної системи та пристроями зберігання даних.

Результатом роботи має стати програмна система, яка надасть користувачу можливість створювати та записувати образи різних типів накопичувачів інформації. Це дозволить ефективно виконувати резервне копіювання, збереження та подальше відновлення даних.

3.1 Опис функціонування системи

Образ диска являє собою спеціальний файл, у якому зберігається повна копія інформації та структура даних певного електронного носія. До таких носіїв можуть належати оптичні диски (CD, DVD, Blu-Ray), розділи жорстких дисків, твердотільні, накопичувачі та інші пристрої збереження даних.

По суті, образ диска містить усі необхідні дані для відтворення вмісту та організації інформації на відповідному носії. Завдяки цьому, його можна використовувати для точного копіювання або відновлення структури й даних пристрою зберігання.

Створення образу може здійснюватися як безпосередньо з фізичного носія, так і на основі використання професійного ПЗ. У більшості випадків формування такого файлу відбувається шляхом посекторного копіювання даних із носія, тобто - дублюванням усіх секторів. При цьому, структура файлової системи зазвичай не аналізується окремо, оскільки вона вже міститься у скопійованих секторах.

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

Основним призначенням образів дисків є забезпечення резервного копіювання даних. Використання цієї технології дає можливість створити повну копію будь-якого носія інформації з подальшою можливістю відновлення його у початковому стані, навіть у разі втрати або пошкодження оригінального носія.

Особливо корисним є створення образу системного розділу комп'ютера або ноутбука. У випадку виникнення збою чи пошкодження операційної системи достатньо виконати відновлення системного розділу з раніше створеного образу, який зберігається у надійному місці. Це дозволяє швидко повернути систему до працездатного стану без необхідності повторного встановлення програмного забезпечення та налаштувань.

Окрім використання для резервного копіювання, образи дисків широко застосовуються і в інших сферах, зокрема серед користувачів комп'ютерних ігор. У такому випадку вони використовуються для підвищення швидкості доступу до даних на зовнішніх носіях інформації.

Читання даних з образу диска, який зберігається на жорсткому диску комп'ютера, відбувається значно швидше, ніж безпосередній доступ до оптичних носіїв, таких, як CD або DVD, через стандартний дисковод. Завдяки цьому забезпечується швидше завантаження гри та більш стабільна її робота під час використання.

Ще одним важливим напрямом застосування резервного копіювання є створення копій мобільних носіїв інформації. До таких пристроїв належать переносні накопичувачі на жорстких дисках, USB-накопичувачі та флеш-носії, які широко використовуються для зберігання та перенесення даних. Особливістю подібних пристроїв є те, що вони можуть виконувати не лише функцію звичайного носія інформації, а й використовуватися як повноцінні завантажувальні носії операційних систем. Наприклад, на флеш-накопичувач або зовнішній жорсткий диск можна встановити операційну систему та здійснювати її запуск безпосередньо з цього пристрою. Крім того, такі носії іноді застосовуються як засоби доступу або ключі безпеки, коли запуск певної програми чи відкриття

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

файлу можливе лише за умови підключення відповідного флеш-драйва до комп'ютера.

У разі виходу з ладу мобільного носія користувач може опинитися у безвихідній ситуації: всі необхідні програми та файли присутні, проте без фізичного ключа їх запуск неможливий. Тому доцільно створювати дублікати таких ключів на аналогічних пристроях, особливо враховуючи відносно низьку вартість сучасних флеш-накопичувачів. Найзручнішим способом є формування образу носія та його копіювання на інший еквівалентний пристрій за допомогою спеціального програмного забезпечення, яке планується розробити в рамках виконання бакалаврської випускної кваліфікаційної роботи.

Відповідно до технічного завдання розробляється автономна система для автоматизованого збереження інформації, проте слід передбачити її інтеграцію у складі більш потужної системи аналогічного типу. Для цього до складу ПЗ треба включити функціональні заглушки, які в майбутньому дозволять підключати нові модулі без суттєвих змін у існуючій системі.

Згідно з методикою побудови систем подібного класу та призначення (розділ 2 пояснювальної записки), система повинна забезпечити два режими роботи:

- автоматичний;
- інтерактивний.

Інтерактивний режим призначений для безпосередньої взаємодії користувача із системою, що дозволяє контролювати процеси та виконувати необхідні дії вручну.

Отже, основними завданнями розробки програмного забезпечення для створення образів, які реалізуються в процесі виконання ВКРБ, є:

- створення та реалізація програмних інструментів для читання та запису даних між цифровими носіями інформації та жорстким диском;
- визначення списку підтримуваних файлових систем і забезпечення можливості їхнього розширення в майбутньому;

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

- розробка механізму створення образів через командний рядок для автоматизації процесів за допомогою планувальників завдань операційної системи.

Визначимо основні джерела вхідних даних для системи, до яких належать:

- фізичні носії інформації, а також каталоги та окремі файли;
- ресурси, доступні через мережу.

Враховуючи зазначене, визначимо вимоги до організаційної структури системи:

- забезпечення швидкого та надійного керування робочими файлами;
- запобігання дублюванню даних та виникненню неконтрольованих ланок і процесів;
- реалізація прийняття рішень лише через центральний компонент системи;
- надання користувачеві достатньої інформаційної підтримки, включаючи розвинений режим допомоги (HELP) та можливості візуалізації процесів.

Задля реалізації операцій читання та запису даних між цифровими носіями інформації та жорстким диском в якості базової функції обирається DeviceIoControl ОС Windows. Використання цієї функції дозволяє організувати прямий доступ до драйверів операційної системи та забезпечує роботу на низькому рівні.

Синтаксис функції DeviceIoControl у поєднанні з API ОС Windows виглядає наступним чином:

```
BOOL DeviceIoControl(  
    HANDLE hDevice,  
    DWORD dwIoControlCode,  
    LPVOID lpInBuffer,  
    DWORD nInBufferSize,  
    LPVOID lpOutBuffer,  
    DWORD nOutBufferSize,  
    LPDWORD lpBytesReturned,  
    LPOVERLAPPED lpOverlapped  
);
```

Функція забезпечує передачу керуючих кодів та буферів даних для виконання низькорівневих операцій з пристроями зберігання інформації.

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

Параметр `hDevice` [in] – дескриптор пристрою, на якому виконується операція. Це може бути том, каталог, файл або потік даних. Для отримання відповідного дескриптора використовується функція `CreateFile`, яка забезпечує доступ до обраного об'єкта в системі.

`dwIoControlCode`

[in] `dwIoControlCode` – код керування операцією. Він визначає конкретну дію, яка повинна бути виконана, а також тип пристрою, на якому ця операція відбувається. Документація для кожного коду містить детальний опис необхідних параметрів і способів їх використання під час виклику функції:

`lpInBuffer`, `nInBufferSize`, `lpOutBuffer` і `nOutBufferSize`.
`lpInBuffer`
[in]

[in] `lpInBuffer` – вказівник на буфер для введення даних, необхідних для реалізації операції. Тип даних задається значенням параметра `dwIoControlCode`. Якщо обрана операція не потребує передачі вхідних даних, параметр прийме значення `NULL`.

`nInBufferSize`
[in] - розмір буфера введення даних, у байтах.
`lpOutBuffer`

[out] `lpOutBuffer` – вказівник на буфер для виведення даних, у який функція записує результати виконання операції. Структура даних регламентується значенням параметра `dwIoControlCode`. Якщо обрана операція не передбачає повернення даних, параметр прийме значення `NULL`.

`nOutBufferSize`
[in] - розмір буфера виведення даних, у байтах.
`lpBytesReturned`

[out] `lpBytesReturned` – вказівник на змінну, у якій функція зберігає кількість байтів даних, записаних у буфер виведення. Якщо розмір буфера виявляється недостатнім для збереження результатів операції, виклик функції завершиться помилкою, функція `GetLastError` поверне код `ERROR_INSUFFICIENT_BUFFER`, а значення параметра `lpBytesReturned` буде рівне нулю.

Якщо буфер для виведення даних занадто малий для розміщення всіх результатів операції, але може містити частину інформації, деякі драйвери повертають лише ті дані, які помістилися. У цьому випадку виклик функції завершиться помилкою, а функція GetLastError поверне код ERROR_MORE_DATA. Значення параметра lpBytesReturned покаже обсяг фактично отриманих даних. Для завершення операції програмний додаток викликає DeviceIoControl повторно з тією ж операцією, встановлюючи нову точку відліку для продовження передачі даних.

Якщо параметр lpOverlapped встановлено в NULL, то lpBytesReturned не може бути порожнім. Навіть у випадку, коли операція не повертає даних і lpOutBuffer дорівнює NULL, функція DeviceIoControl використовує lpBytesReturned, хоча після завершення операції його значення вже не має практичного сенсу.

Якщо ж lpOverlapped не дорівнює NULL, то lpBytesReturned може бути порожнім. При цьому, якщо асинхронна операція повертає дані, значення lpBytesReturned стає актуальним лише після завершення цієї операції.

Задля отримання інформації про кількість фактично повернутих байтів у асинхронній операції використовується функція GetOverlappedResult. Якщо дескриптор пристрою hDevice пов'язаний з портом завершення введення/виведення (I/O), кількість оброблених даних можна визначити за допомогою виклику GetQueuedCompletionStatus.

lpOverlapped

[in] - покажчик на структуру OVERLAPPED.

Якщо hDevice відкрито без прапорця FILE_FLAG_OVERLAPPED, параметр lpOverlapped ігнорується.

Якщо ж дескриптор hDevice відкрито з FILE_FLAG_OVERLAPPED, операція виконується асинхронно. У цьому випадку lpOverlapped має вказувати на коректну структуру OVERLAPPED, яка містить дескриптор об'єкта події; інакше виклик функції завершиться помилкою.

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

У випадку асинхронних операцій функція DeviceIoControl повертає управління одразу, а відповідний об'єкт події сигналізує про завершення операції. Якщо ж операція синхронна, функція не повертає значення до моменту її завершення або виникнення помилки.

Можливі результати виконання функції:

- при успішному завершенні DeviceIoControl повертає ненульове значення;
- при помилці функція повертає 0, а додаткову інформацію про причину можна отримати викликом GetLastError.

Щоб витягнути дані про дескриптор пристрою, необхідно викликати функцію CreateFile або з ім'ям пристрою, або з ім'ям драйвера, пов'язаного з пристроєм. Щоб визначити ім'я пристрою, використаємо нижче перерахований формат:

\\.\DeviceName

Функція DeviceIoControl приймає дескриптор конкретного пристрою.

При виклику CreateFile для відкриття дескриптора драйвера пристрою, слід використовувати прапорці доступу FILE_SHARE_READ і FILE_SHARE_WRITE. Для комунікаційних ресурсів, таких як послідовний порт, необхідно забезпечити монопольний доступ. Інші параметри виклику CreateFile для відкриття дескриптора пристрою наведено нижче:

- fdwCreate слід встановити як OPEN_EXISTING;
- hTemplateFile має бути NULL;
- fdwAttrsAndFlags можна задати як FILE_FLAG_OVERLAPPED, щоб дескриптор підтримував асинхронні операції введення-виведення (I/O).

Перелік керуючих кодів, які буде підтримувати розроблювана система:

- коди для роботи з комунікаційними пристроями;
- коди для управління апаратними пристроями;
- коди для керування каталогами;
- коди для роботи з дисками;

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

- коди для управління файлами;
- коди для взаємодії з файловими системами;
- коди для керування живленням комп'ютера;
- коди для роботи з томами.

Файлова система виконує роль посередника між фізичним носієм інформації та АРІ для роботи з файлами. Коли прикладна програма звертається до файла, їй не потрібно знати, як саме організовані дані всередині файла або на якому типі носія вони зберігаються - це може бути CD, жорсткий диск, магнітна стрічка, флеш-накопичувач або інший пристрій. Програма отримує лише базову інформацію: ім'я файла, його розмір та атрибути, які надає драйвер файлової системи. Відповідно, саме файлова система визначає, де і яким чином дані будуть розміщені на фізичному носії.

З точки зору ОС диск представляє собою набір кластерів, зазвичай розміром від 512 байт і більше. Драйвери файлової системи відповідають за організацію цих кластерів у файли та каталоги. Отже, у такому випадку каталог реально є файлом, котрий утримує список файлів, які він включає. Крім того, драйвери контролюють стан кластерів, визначаючи: які з них зайняті, які вільні, а також помічені як пошкоджені чи несправні.

Однак файлова система не обов'язково повинна бути прив'язана до конкретного фізичного носія. Існують віртуальні файлові системи, а також мережеві файлові системи, які забезпечують доступ до файлів, розташованих на віддаленому комп'ютері, виступаючи лише як інтерфейс для роботи з ними.

Розроблюване програмне забезпечення буде сумісне з сучасними та широко використовуваними файловими системами, включаючи: exFAT, FAT32, NTFS, ISO 9660, Ext2/3/4. Це дозволяє працювати з більшістю сучасних операційних систем і типів носіїв інформації.

Файлові системи exFAT і FAT32.

Файлова система exFAT та її попередник FAT32 (від англ. File Allocation Table - таблиця розміщення файлів) походять від 16-розрядної системи FAT16, яка

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

використовувалася в ранніх версіях MS-DOS і Windows. FAT32 була впроваджена для підтримки жорстких дисків великого об'єму та інших покращених функцій у сучасних версіях Windows, таких як Windows 2000, Windows XP і пізніші. У подальшому під терміном FAT розуміють будь-яку з цих версій.

На сьогодні FAT32 використовується переважно для знімних носіїв та сумісності між різними операційними системами, тоді як exFAT забезпечує підтримку великих обсягів даних і сучасних накопичувачів, включаючи флеш-накопичувачі та зовнішні SSD.

NTFS (New Technology File System) є сучасною файловою системою (ФС), що застосовується у всіх сучасних версіях Windows (від Windows XP до Windows 11). Вона підтримує великі обсяги даних, розширені атрибути файлів, контроль доступу, журнальну файлову систему для підвищення надійності, а також - шифрування та стиснення файлів.

ISO 9660 та Ext2/3/4.

ISO 9660 використовується для роботи з оптичними дисками (CD/DVD/Blu-Ray).

Файлові системи Ext2, Ext3 та Ext4 застосовуються у Linux та Unix-подібних операційних системах і дозволяють ефективно працювати з великими дисковими томами та каталогами.

Завдяки підтримці цих ФС розроблюване ПЗ зможе коректно обробляти дані на різних носіях, забезпечуючи сумісність із сучасними комп'ютерними платформами та міжплатформенний обмін даними.

Файлові системи Ext2, Ext3 та Ext4 застосовуються в операційних системах Linux. Платформи на базі Linux займають кардинальні позиції на світових ринках комп'ютерної техніки, мобільних пристроїв, комп'ютерних технологій та програмних продуктів. Визначимо підхід до реалізації функції ідентифікації статусу ФС.

Функція реалізується у вигляді вкладеної структури if-операторів, де в кожному блоці послідовно виконується перевірка на визначену користувачем ФС.

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

У разі успішного визначення функція повертає результат, а якщо тип файлової системи не визначено, видається відповідне повідомлення про невідому систему:

```
function TSystemFile.GetFSType(): string;
begin
  Result := '';
  if isISO9660() then
    Result := 'ISO 9660'
  else if isExFAT() then
    Result := 'exFAT'
  else if isFAT32() then
    Result := 'FAT32'
  else if isNTFS() then
    Result := 'NTFS'
  else if isExt2_3_4() then
    Result := 'Ext2/3/4'
  else
    Result := 'Unknown';
end;
```

У склад розроблюваного програмного забезпечення слід включити функцію перевірки файлової системи ISO 9660. Вона аналізує вхідні дані носія і повертає True, якщо файлової системі відповідає формат ISO 9660, або False, якщо відповідності немає.

```
function TSystemFile.isISO9660(): Boolean;
var
  ReadStart: _Large_Integer;
  ActualLengthRead: DWORD;
  Seek: DWORD;
  Buffer: array[0..7] of Char;
begin
  Result := False;
  ReadStart.QuadPart := $8000; // початкова позиція для перевірки ISO 9660

  // Переміщення вказівника файлу
  Seek := SetFilePointer(FileHandle, ReadStart.LowPart, @ReadStart.HighPart,
FILE_BEGIN);
  if Seek <> $FFFFFFFF then
    begin
      // Читання 8 байт для перевірки сигнатури
      if ReadFile(FileHandle, Buffer, 8, ActualLengthRead, nil) then
        begin
          // Перевірка сигнатури ISO 9660
          if (Buffer[0] = #1) and (Buffer[1] = 'C') and (Buffer[2] = 'D') and
(Buffer[3] = '0') and
            (Buffer[4] = '0') and (Buffer[5] = '1') and (Buffer[6] = #1) and
(Buffer[7] = #0) then
            begin
              Result := True;
              SizeOfLabel := 31;
              PositionOfLabel.QuadPart := $8028;
            end;
          end;
        end;
      end;
    end;
```

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

Функція здійснює ідентифікацію файлової системи FAT на носії. Після перевірки функція повертає True, якщо структура файлової системи відповідає FAT, і False, якщо ні.

```
function TSystemFile.isFAT(): Boolean;
var
  ReadStart: _Large_Integer;
  ActualLengthRead: DWORD;
  Seek: DWORD;
  Buffer: array[0..7] of Char; // для читання сигнатури
begin
  Result := False;
  ReadStart.QuadPart := $3; // офсет для перевірки FAT (типово байт 3 для
  OEM name)

  // Переміщення вказівника файлу на початок розділу
  Seek := SetFilePointer(FileHandle, ReadStart.LowPart, @ReadStart.HighPart,
  FILE_BEGIN);
  if Seek <> $FFFFFFFF then
  begin
    // Читання 8 байт для перевірки сигнатури FAT
    if ReadFile(FileHandle, Buffer, 8, ActualLengthRead, nil) then
    begin
      // Перевірка сигнатури FAT (OEM Name містить "MSDOS" або "MSWIN")
      if (Copy(Buffer, 1, 5) = 'MSDOS') or (Copy(Buffer, 1, 5) = 'MSWIN')
      then
      begin
        Result := True;
      end;
    end;
  end;
end;
```

Синтаксис команди для запису файла:

```
rawwritewin [--password] [--write] [--copies n] [--drive driveno] [--yes]
file.img
- password – встановлює пароль для операції;
- write – виконує команду запису файлу на пристрій;
- copies n – визначає кількість копій n, які потрібно записати;
- drive driveno – вказує фізичний номер або літеру логічного диска;
- yes – автоматично підтверджує всі запити;
- file.img – шлях до образу файлу для запису.
```

Синтаксис команди для читання файлу:

```
rawwritewin --read [--drive driveno] file.img
- read – команда для зчитування образу з пристрою;
- drive driveno – фізичний або логічний диск для читання;
file.img – файл, у який буде записано дані.
```

Якщо файла file.img немає, він буде створений, якщо він є – буде перезаписаний зверху наявної інформації або дані будуть дописані до нього (за

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

вибором користувача).

Структурування даних від системних файлів - загальноприйнятий порядок (звичай) серед професіональних користувачів та системних адміністраторів. Для рядових користувачів та приватних осіб - це поділ ЖД на кілька логічних розділів:

- один для операційної системи;
- один для програм;
- один для власних даних.

У корпоративному середовищі великі обсяги інформації зазвичай розташовуються на окремих фізичних дисках, не пов'язаних із системою. Такий підхід спрощує як резервне копіювання, так і зберігання інформації. Принцип роздільного зберігання даних поширюється і на файлові архіви та образи дисків. Їх слід розміщувати хоча б на несистемних розділах того ж жорсткого диска.

Для корпоративних користувачів цього принципу потрібно дотримуватися ще суворіше, як мінімум одна копія даних повинна зберігатися в іншому офісі, філії або відділенні. Це дозволяє уникнути втрати корпоративної інформації через надзвичайні події, такі як повінь, землетрус, пожежа або крадіжка.

3.2 Розробка структурної схеми

Розроблюване програмне забезпечення повинно бути компактним, інтуїтивно зрозумілим та легким для користувача. Воно має надавати необхідну теоретичну підтримку під час роботи із системою. Для цього до складу ПЗ слід включити зручний графічний інтерфейс українською мовою та забезпечити зрозумілі й наочні формати відображення результатів на екрані монітора.

Щоб програмне забезпечення працювало повноцінно, необхідна наступна мінімальна конфігурація:

- комп'ютер Intel® Celeron/8 Mb/1.2 Gb/SVGA 14" 1Mb або сумісні з ним;
- операційна система сучасного покоління Windows.

Образ будь-якого електронного носія інформації (ЕНІ) являє собою файл-

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

контейнер, який є точною побітовою копією (дзеркальним відображенням) фізичного носія інформації: CD/DVD-диска, флешки або розділу ЖД. Він зберігає не лише файли, а й структуру файлової системи, завантажувальні записи та параметри носія. Іншими словами, такий образ зберігає всю інформацію, необхідну для повного відтворення розташування і доступу до даних на будь-якому носії. Зазвичай, при створенні образу ЕНІ, який відрізняється за типом, відтворюються сектори носія, тоді як існуюча на ньому файлова система часто ігнорується.

Під час розробки ПЗ ВКРБ планується використовувати нестандартний підхід до створення образів ЕНІ. На відміну від традиційного методу, який ігнорує файлову систему джерела, система повинна забезпечувати створення образу з урахуванням його файлової системи.

Спочатку образи дисків застосовувалися головним чином для резервного дублювання даних, де точне відтворення структури вихідного носія було критично важливим. З появою оптичних носіїв (CD, DVD) образи цих дисків набули більшого поширення, переважно у форматі *.ISO, який зберігає файлову систему ISO 9660, типовою для CD і DVD. Однак цей формат має обмеження. Він не підтримує багатосесійні диски та аудіо-CD.

Окрім формату ISO, на сьогодні існує безліч інших форматів образів дисків, таких як *.IMG і *.DMG, а також пропрієтарні формати: *.NRG (Nero Burning ROM), *.MDS/ *.MDF (DAEMON Tools, Alcohol 120%), *.DAA (PowerISO), *.PQI (DriveImage) та ще - *.CCD/ *.IMG/ *.SUB (CloneCD). Найбільш універсальним форматом для електронних носіїв інформації (ЕНІ) вважається *.IMG. Саме на нього буде орієнтована розробка системи для створення образів ЕНІ.

У основу програмного забезпечення майбутньої системи закладається модульний принцип побудови: головний керуючий модуль та допоміжні підмодулі, кожен із яких виконує одну конкретну функцію.

Такий підхід дозволяє розробнику легко модифікувати окремий модуль, усувати помилки, додавати нові функції (наприклад, додаткові операції) та без

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

проблем передавати оновлений модуль користувачу через ЕНІ або мережу Internet. Користувачу достатньо скопіювати новий модуль у директорію програми, і пакет автоматично оновлюється. Перезавантаження системи не потрібне - можна відразу запускати програму та виконувати необхідні операції.

При розробці концепції майбутньої системи та шляхів її реалізації основний акцент був зроблений на швидкодії, зрозумілості та простоті у роботі з електронними носіями інформації, що забезпечує легкий і зручний доступ до всіх функцій системи.

Отже, у процесі виконання ВКРБ передбачається розробка складної багатофункціональної системи, для реалізації якої буде використано сучасні Windows Runtime (WinRT) API та компоненти сучасних версій Windows. WinRT надає програмам можливість безпосередньо викликати системні служби та підпрограми ОС, одночасно забезпечуючи кращу інтеграцію з сучасними графічними інтерфейсами та апаратними можливостями.

При створенні ПЗ системи будуть задіяні такі основні бібліотеки та модулі:

- Windows.Foundation та Windows.System - забезпечують управління пам'яттю, процесами, задачами та іншими ресурсами системи;

- Windows.UI.Xaml - надає засоби для розробки сучасного, зручного графічного інтерфейсу користувача;

- Windows.Security.Credentials та Windows.Storage - містять функції для доступу до системного сховища та управління безпекою даних.

Крім цього, для реалізації повного функціоналу необхідно враховувати системні інтерфейси ядра та драйверів:

- бібліотеку драйверів пристроїв, для яких будуть створюватися образи;

- Native API ядра (ntoskrnl.exe);

- рівень Hardware Abstraction Layer (hal.dll) для взаємодії з апаратурою.

Такий підхід дозволяє досягти високого рівня продуктивності, безпосереднього контролю за ресурсами та повної функціональної сумісності з сучасними версіями Windows, зберігаючи при цьому модульність і

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

розширюваність системи.

Використання зазначених компонентів дозволяє організувати ефективну взаємодію між пристроями та програмним забезпеченням системи, що необхідно для реалізації головного завдання - створення електронних образів пристроїв. На рисунку 3.1 наведено структуру взаємозв'язків ПЗ з ЕНІ на основі сучасних API Windows при формуванні образів. Розроблену структуру будемо використовувати як базову для побудови архітектури майбутньої системи. Детальніше охарактеризуємо основні компоненти та їх функції:

- HAL.dll (Hardware Abstraction Layer) - рівень абстракції від апаратного забезпечення. Використання цього компонента забезпечує платформну незалежність системи.

- NTdll.dll - виступає як «місток» між ядром ОС та бібліотеками, які працюватимуть у складі системи, забезпечуючи виклики Native API сучасного ядра Windows.

- Графічна підсистема та бібліотеки Windows.UI.Xaml / Direct2D - ці сучасні компоненти забезпечують чітке і стабільне відображення графічного інтерфейсу користувача, надаючи програмам примітиви для малювання вікон та елементів керування.

- Функціонал керування пристроями (аналог DeviceIoControl через Windows.Devices API) - дозволяє ініціювати виконання обраних операцій на пристрої. Для визначення типу підключеного пристрою використовується сучасний DeviceInformation / GetDriveType API.

При створенні образу диска необхідно заздалегідь визначити його основні параметри, серед яких: обсяг вільного місця в байтах; загальна ємність накопичувача в байтах; інформація про файлову систему, розташовану на обраному носії (за потреби); назва диска; кореневий каталог; серійний номер диска (службовий).

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

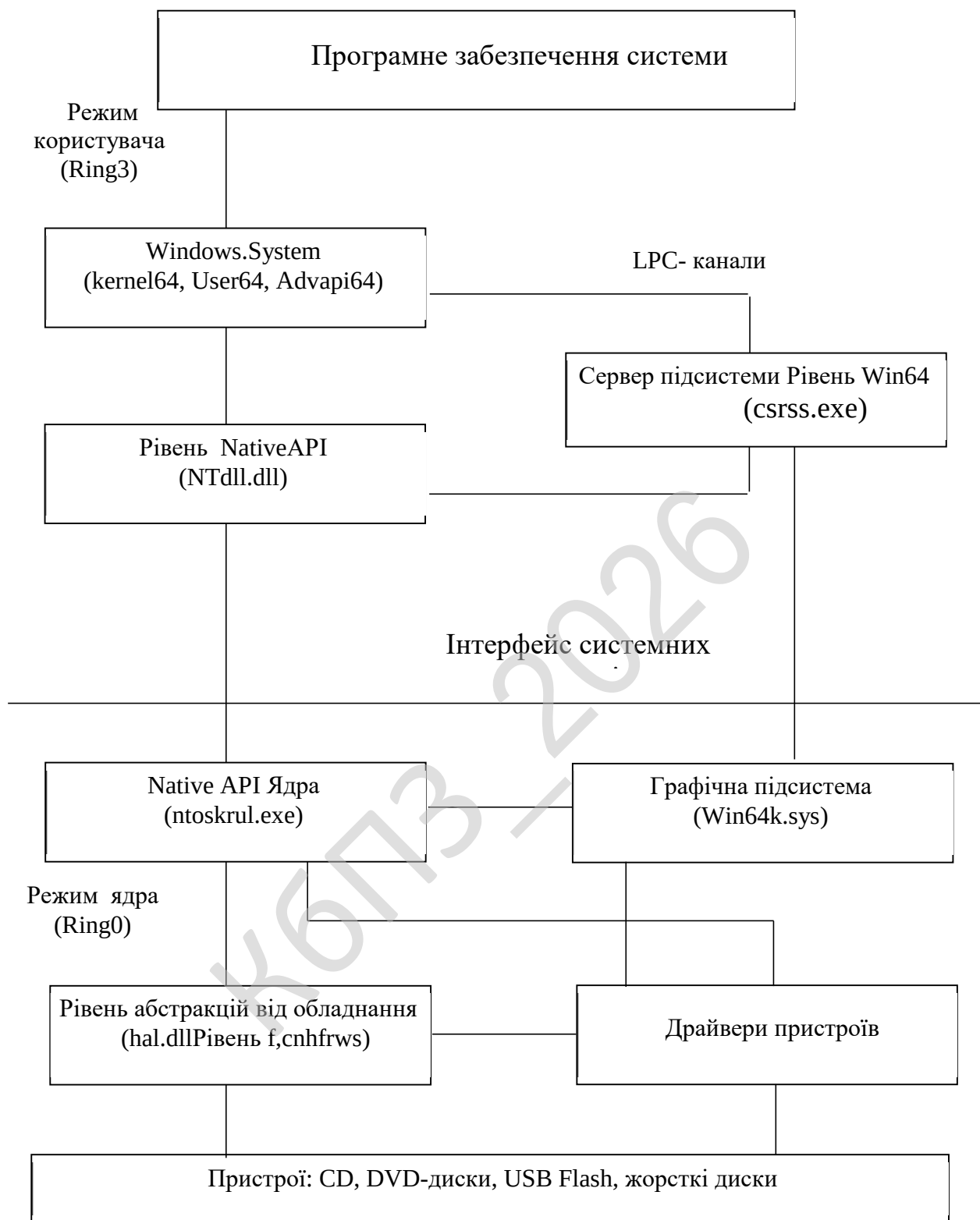


Рисунок 3.1 – Структура взаємозв'язків ПЗ системи з пристроями при створенні їх образів

Як було означено в підрозділах 2.2 та 3.1 цієї пояснювальної записки, модульна структура програмної системи - це метод проектування, який передбачає розбиття складної програми на окремі, функціонально завершені блоки (модулі). Кожен модуль буде реалізовувати в розробленій автоматизованій системі конкретну функцію, бо має чіткий інтерфейс, інкапсулює внутрішню логіку та може розроблятися й тестуватися незалежно від інших компонентів, що підвищує якість коду.

Модульність дозволить використати при проектуванні нашої системи ієрархічну структуру ПЗ. Це значно полегшить не тільки сам процес розробки ПЗ, а й в подальшому забезпечить підтримку та розширення програмного забезпечення при експлуатації у Замовника. До складу системи введемо такі функціональні модулі: головна програма, читання образів, запису образів.

Головна програма (графічний інтерфейс) - це точка входу та керуючий центр. Вона буде ініціалізувати компоненти, керувати логікою виконання, викликати в разі необхідності інші підпрограми, обробляти дані та завершувати роботу. Її основна мета - забезпечення цілісного функціонування системи, об'єднуючи окремі модулі в єдине ціле.

Функція контролю та спостереження (моніторинг) при роботі системи забезпечить стабільність, вчасне виявлення помилок та ефективність роботи. Вона дозволить автоматизувати збір даних про роботу додатків, аналізувати продуктивність, попереджати збої та забезпечувати зворотний зв'язок для оптимізації. При програмній реалізації функції контролю використаємо технологію API ОС Windows.

Моніторинг дозволить виявляти критичні помилки та відхилення від планових показників у режимі реального часу, запобігаючи серйозним збоєм; відстежувати використання ресурсів та дій користувачів, що мінімізує ризики безпеки; забезпечить оцінку того, чи досягає система запланованих цілей.

Переходимо до розробки структурної схеми системи (рисунк 3.2).

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

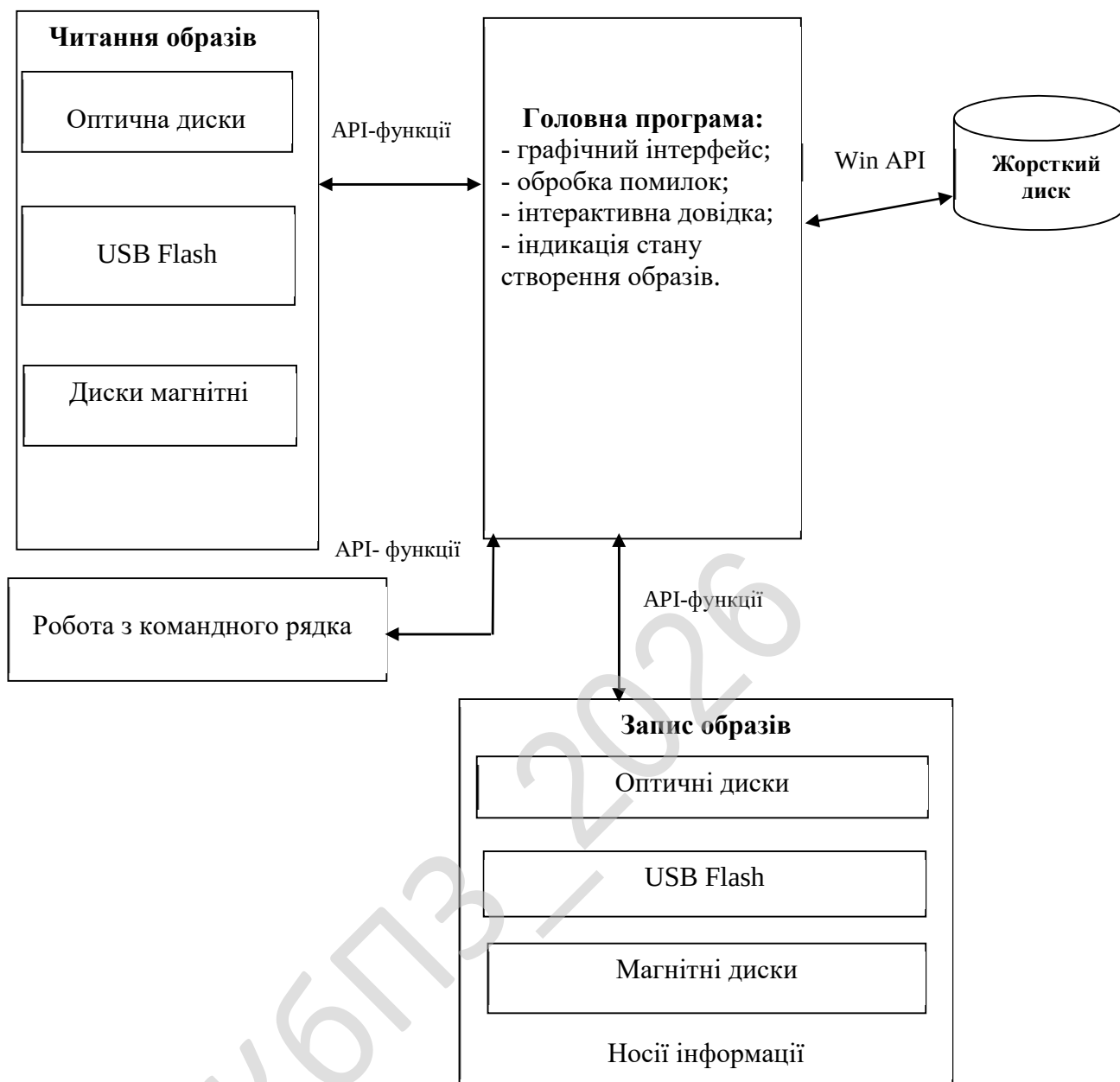


Рисунок 3.2 – Структурна схема системи

3.3 Розробка функціональної схеми

Для побудови функціональної схеми системи необхідно конкретизувати функції кожного компоненту системи та визначити взаємозв'язок між ними.

До складу системи, що підлягає розробці, слід ввести два інтерфейси для забезпечення двох режимів роботи:

- Інтерактивного: керуючі впливи надає користувач в залежності від бажання почати/продовжити/примусово завершити роботу системи. Для інтерактивного режиму буде розроблено графічний інтерфейс.

- Автоматичного: керуючі впливи надходять через командний рядок та інтерфейс, передбачений для керування системою з зовнішнього середовища. Наприклад, від іншої програми.

Система дозволяє створювати електронні образи будь-якого типу носія інформації, якщо він попередньо в системі вже зареєстрований: оптичних дисків будь-якої ємності; CD, DVD, HD, BR-диски; USB Flash будь-якої конфігурації; жорсткий диск.

Образ носія інформації зберігається на попередньо заданому жорсткому диску у вигляді файла з розширенням *.img. Запис образів дисків відбувається без компресії. Проте, щоб успішно конкурувати з наявними на ринку програмних продуктів системами-аналогами, необхідно, щоб система забезпечувала виконання (окрім функцій по безпосередньому призначенню) наступних функцій:

- висока швидкість створення образу електронного носія інформації;
- мати україномовний та підсвідомо доступний інтерфейс користувача, який не вимагає від нього спеціальних знань з основ програмування;
- можливість запису з образу на інший носій інформації;
- мати невеликий обсяг програмного коду;
- вимагати для роботи мінімальну архітектуру програмно-апаратного комплексу.

На рисунку 3.3 наводимо функціональну схеми системи.

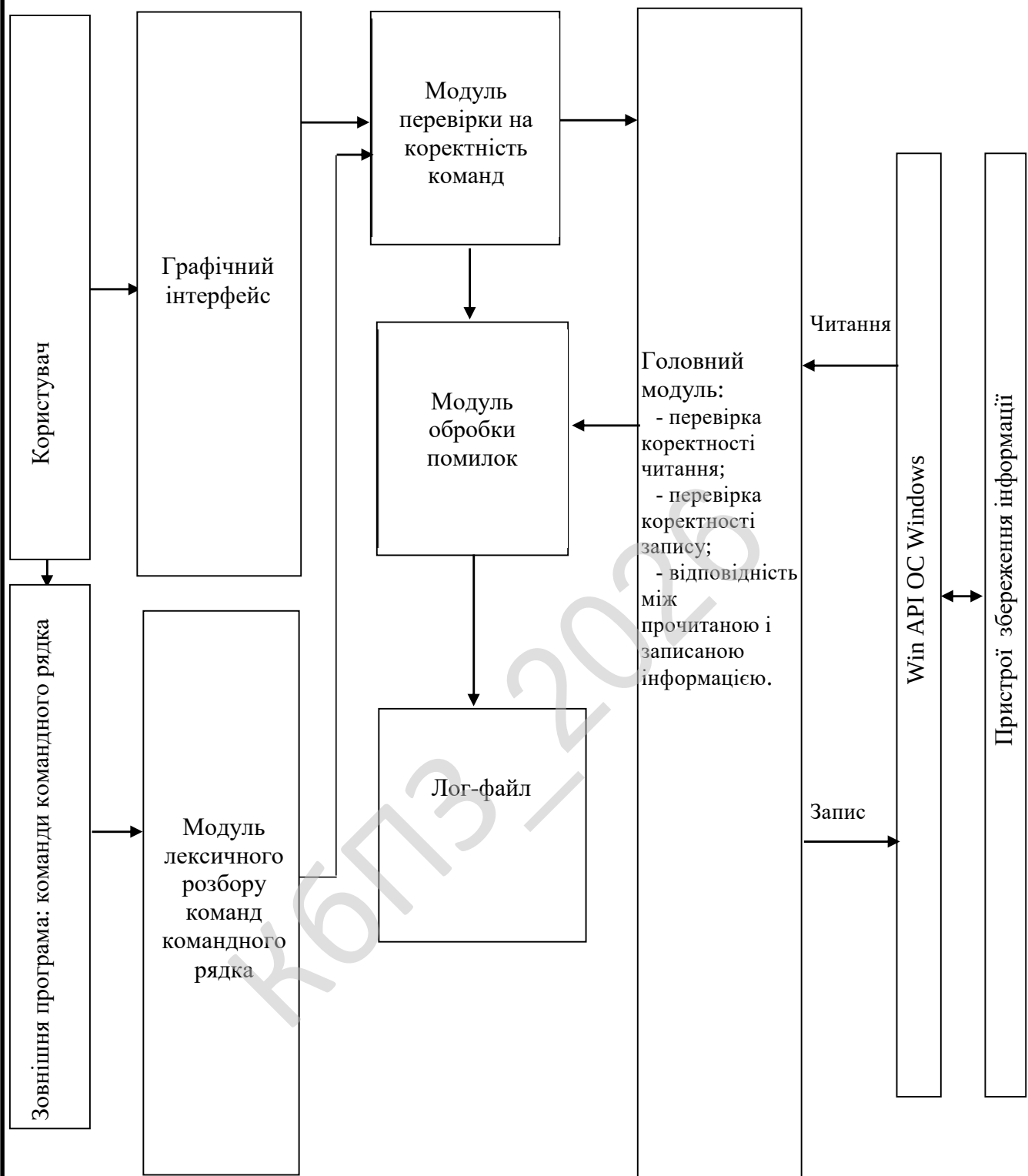


Рисунок 3.3 - Функціональна схема системи

3.4 Розробка діаграми процесів

Для визначення ієрархії процесів та їх взаємодії в системі розроблена та наведена на рисунку 3.4 діаграма взаємодії процесів в системі.

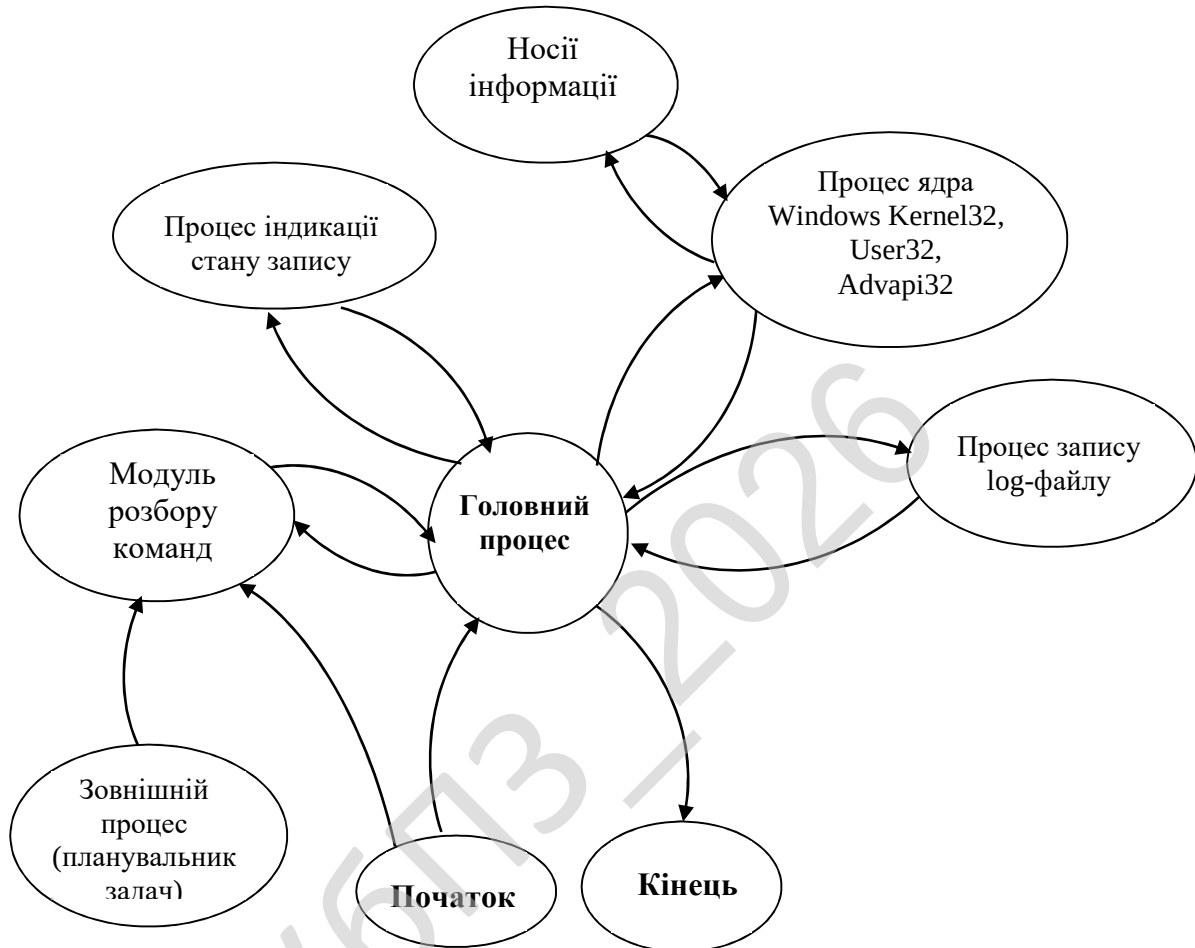


Рисунок 3.4 - Діаграма взаємодії процесів в системі

Головний процес системи завантажується в двох режимах:

- користувачем за допомогою графічного інтерфейсу в інтерактивному режимі;
- з командного рядка через планувальник задач чи іншу зовнішню програму в автоматичному режимі.

На початок роботи системи головний процес через графічний інтерфейс запускає процес індикації стану читання/запису на графічний формі та процес запису/візуалізації службової інформації про помилки в лог-файл.

Головний процес, використовуючи API-функції, через ядро ОС Windows: Kernel64, User64, Advapi 64, керує операціями читання/запису/примусової зупинки носіїв інформації.

При роботі через зовнішню програму (планувальник задач) запускається процес розбірки команд, а далі – алгоритм роботи, аналогічний функціонуванню графічного інтерфейса.

Таким чином, фактично завершена теоретична побудова системи, оскільки визначені та обґрунтовані:

- основні принципи та методика побудови системи;
- проведено опис функціонування системи;
- визначені основні складові системи та проведено обґрунтування цього вибору;
- описаний хід теоретичної побудови системи, розроблена модель проекту: структурна схема, функціональна схема та діаграма процесів.

Маємо всі необхідні дані для виконання основної задачі по реалізації системи – розробки програмного забезпечення.

4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ

4.1 Розробка блок-схем та опис алгоритмів функціонування системи

Відповідно до вимог ТЗ та постановки задачі метою виконання ВКРБ є розробка ПЗ системи, яка будучи встановленою на ПК користувача, забезпечить створення/збереження/запис на інші носії інформації образу ЕНІ.

Образи дисків - це файли, в яких утримується повна інформація в необробленому вигляді (формат *.raw). Образи дисків з форматом *.img не можна скопіювати на інший електронний носій інформації звичним чином. Для цього необхідно використати спеціально розроблену програму, яка буде виконувати цю операцію в нестандартному режимі. Адже образи представляють собою необроблені дані електронного накопичувача, для запису яких необхідно виконання посекторового копіювання даних із файла на інший накопичувач. Саме ці задачі і підлягали, згідно ТЗ, програмній реалізації в процесі виконання ВКРБ.

Для розширення функціоналу системи до її складу введемо два режими роботи:

- створення та збереження у вигляді образів носіїв інформації: жорсткий диск, USB Flash та оптичні диски;
- копіювання створеного образу ЕНІ на будь-який інший носій інформації після його збереження на жорсткий диск.

При побудові архітектури ПЗ було вирішено розробити два взаємопов'язані модулі:

- unit rawwrite: основна реалізація функцій системи: робота графічного інтерфейса (інтерактивний режим роботи користувача), організація автоматичного режиму роботи системи за допомогою планувальника задач, обробка зовнішніх

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

задач (від інших програм/систем), визначення переліку підключених ЕНІ, керування процедурами читання/запису/створення образу ЕНІ та його копіювання;

- unit BlockDev: читання/запис/копіювання інформації ЕНІ, які підлягають обробці в системі з метою створення їх образу.

Це відповідає методологічним підходам до процесу проектування програмних систем та ґрунтується на концептуальних засадах їх розробки.

Означимо основні кроки структури загального алгоритму системи:

- Крок 1: початкове налаштування структури вибраного (по замовченню) пристрою з випадального списку (автоматичний режим - роботи системи).

- Крок 2: одержання хендлера файла пристрою, встановлення над ним контролю для прямої роботи з драйвером.

- Крок 3: визначення файлової структури обраного пристрою;

- Крок 4: читання файлової структури: та збереження створеного образу пристрою на жорсткий диск.

- Крок 5: копіювання з жорсткого диску образу у ЕНІ (пристрою).

- Крок 6: робота користувача з системою (робота графічного інтерфейсу).

- Крок 7: вибирання джерела створення образу, ініціалізація його структури.

- Крок 8: повторення кроків 2, 3, 4, 5.

На кожному кроці роботи алгоритму буде передбачена розвинена мережа автоматичної перевірки роботи функцій та процедур (стандартних та розроблених автором ВКРБ), яка забезпечить надійну та коректну роботу системи. Наявність в системі графічного інтерфейса забезпечить користувача достатньою ступінню інформативності. Окрім цього, в системі передбачена індикація процесу створення образу.

Розглянемо більш детально кожний крок розробленої структури алгоритму та визначимо шляхи його програмної реалізації.

Синхронний прийом/передачу даних організуємо в системі, використавши функції ReadFile і WriteFile. Означимо синтаксис цих функцій:

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

```
function ReadFile (hFile: THandle; var Buffer;
    nNumberOfBytesToRead: DWORD;
    var lpNumberOfBytesRead: DWORD;
    lpOverlapped: POverlapped): BOOL;
function WriteFile (hFile: THandle; const Buffer;
    nNumberOfBytesToWrite: DWORD;
    var lpNumberOfBytesWritten: DWORD;
    lpOverlapped: POverlapped): BOOL;
```

Параметр hFile – дескриптор порта; параметр Buffer – буфер, дані з якого передаються чи приймаються.

Параметри nNumberOfBytesToRead та nNumberOfBytesToWrite – число байт, які повинні бути передані/прийняті.

Параметри lpNumberOfBytesRead та lpNumberOfBytesWritten – число реально переданих/прийнятих байт. Це число може бути менше очікуваного із-за помилок та переривання операції читання/запису по тайм-ауту.

Оскільки будемо використовувати в системі тільки синхронний прийом/передачу даних, то параметр lpOverlapped буде мати значення nil.

На початку роботи системи по створенню ЕНІ необхідно одержати різнобічну інформацію про об'єкт файлової системи: файл, теку, диск та одержати доступ до списку піктограм Windows. Для цього використаємо функцію SHGetFileInfo(pszPath: PansiChar, яка має наступний синтаксис:

```
function SHGetFileInfo(pszPath: PansiChar;
    dwFileAttributes: DWORD;
    var psfi: TSHFileInfo;
    cbFileInfo, uFlags: UINT): DWORD;
```

Параметр pszPath задає файл чи шаблон файлів, про які необхідно одержати інформацію.

Параметр dwFileAttributes визначає атрибути файла і буде використовуватись лише при наявності прапорця:

SHGFI_USEFILEATTRIBUTES:

В інших випадках будемо привласнювати йому значення nil.

В запис Psfi, який передається по посиланню, функція занесе одержану інформацію, а параметр cbFileInfo задасть розмір запису Psfi.

Запис Psfi утримує наступні поля:

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

поле	тип
nIcon	HICON
iIcon	Integer
dwAttributes	DWORD
szDisplayName	array [0..MAX_PATH-1] of AnsiChar или WideChar
szTypeName	array [0..79] of AnsiChar или WideChar

Параметр uFlags буде включати наступні прапорці:

- SHGFI_ATTRIBUTES: Одержання атрибутів файла. Прапорці атрибутів заносяться в поле dwAttributes запису psfi.
- SHGFI_DISPLAYNAME: Одержання повного імені файла. Ім'я заносяться в поле szDisplayName запису psfi.
- SHGFI_EXETYPE: Якщо pszPath - ім'я виконуємого файла, то функція повертає тип файла.
- SHGFI_ICON: Доступ до дескриптора піктограми файла чи до системного списку піктограм. Використовується з рядом модифікаторів. Дескриптор заносяться в поле hIcon запису psfi, а індекс піктограми - в поле iIcon. Функція повертає дескриптор системного списку піктограм.
- SHGFI_ICONLOCATION: Доступ до імені файла, утримуючого піктограму даного файла. Ім'я заносяться в поле szDisplayName запису psfi.
- SHGFI_LARGEICON: Модифікатор SHGFI_ICON, надає доступ до списку великих піктограм.
- SHGFI_LINKOVERLAY: Модифікатор SHGFI_ICON, надає оверлейний зв'язок з піктограмами файлів.
- SHGFI_OPENICON: Модифікатор SHGFI_ICON, надає доступ до списку піктограм, відповідних відкритому контейнеру (теці, каталогу).
- SHGFI_PIDL: Вказує, що pszPath - адреса структури ITEMIDLIST, а не ім'я файла.
- SHGFI_SELECTED: Модифікатор SHGFI_ICON, який надає доступ до списку піктограм з кольором, відповідним означенню.
- SHGFI_SHELLICONSIZE: Модифікатор SHGFI_ICON, надає доступ до списку піктограм, розмір яких визначається середовищем.

- SHGFI_SMALLICON: Модифікатор SHGFI_ICON, надає доступ до списку малих піктограм.

- SHGFI_SYSICONINDEX: Доступ до індексу піктограми в системному списку. Індекс заноситься в поле iIcon запису psfi. Функція повертає дескриптор системного списку піктограм.

- SHGFI_TYPENAME : Доступ до рядка опису типу файла. Рядок заноситься в поле szTypeName запису psfi.

- SHGFI_USEFILEATTRIBUTES: Вказує, що функція повинна використати параметр dwFileAttributes.

Виконуваний файл - це бінарний файл, що містить машинні інструкції (код), які безпосередньо виконуються процесором, або скрипти, що інтерпретуються операційною системою як програма. Виконувані файли відрізняються від файлів даних (документів, зображень), оскільки вони не просто відкриваються іншою програмою, а самі є програмою, що виконує дії.

При визначенні типу виконуваного файлу, в параметрі uFlags задаємо тільки SHGFI_EXETYPE. Тоді повернуте функцією SHGetFileInfo значення вкаже тип файлу:

```
0 // файл не підлягає виконанню/помилка;  
LOWORD = NE або PE (17744) HIWORD = 3.0, 3.5 чи 4.0 додаток Windows;  
LOWORD = MZ (23117) HIWORD = 0 //файли MS-DOS .EXE, .COM чи .BAT  
LOWORD = PE (17744) HIWORD = 0 //консольний додаток Win32
```

Призначення функції SHGetFileInfo означає доступ до системних списків піктограм. Сполучення прапорців SHGFI_SMALLICON or SHGFI_ICON or SHGFI_SYSICONINDEX або SHGFI_LARGEICON or SHGFI_ICON or SHGFI_SYSICONINDEX надає доступ до списків малих і великих піктограм Windows.

Список усіх доступних ЕНІ одержимо, використавши функцію GetLogicalDriveStrings, яка об'явлена в модулі Windows наступним чином:

```
function GetLogicalDriveStrings(nBufferLength: DWORD;  
lpBuffer: PAnsiChar): DWORD; stdcall;
```

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

Параметр `nBufferLength` визначає довжину буфера, в який функція заносить список ЕНІ. Вона поверне довжину буфера, яка необхідна для розміщення всієї інформації.

Якщо повернута довжина є більшою, чи `nBufferLength`, значить, список не вдалось занести повністю. Якщо функція повернула 0, значить виникла помилка.

Список розташуємо в буфері в вигляді рядків, які означають кореневі каталоги, наприклад: «C:\». Окремі рядки відокремимо один від одного нульовими символами. В кінець списку занесемо два нульових символи.

Для визначення типу конкретного ЕНІ використаємо функцію `GetDriveType`:

```
function GetDriveType(lpRootPathName: PChar): UINT; stdcall;
```

Параметр `lpRootPathName` – це кореневий каталог ЕНІ в тій формі, в яку заносить ЕНІ в список функція `GetLogicalDriveStrings`. Якщо `lpRootPathName=nil`, то це поточний ЕНІ. Функція повертає ціле число, яке може приймати наступні значення:

- 0 – тип диску не визначений;
- 1 – заданий кореневий каталог відсутній;
- `DRIVE_REMOVABLE` – з'ємний диск;
- `DRIVE_FIXED` – постійний диск;
- `DRIVE_REMOTE` – віддалений мережевий диск;
- `DRIVE_CDROM` – диск CD-ROM;
- `DRIVE_RAMDISK` – віртуальний диск в пам'яті.

Інформацію про розміри диска/ЕНІ та наявності вільного місця на ньому будемо одержувати, використавши функцію:

```
GetDiskFreeSpaceEx, яку об'явимо наступним чином:  
function GetDiskFreeSpaceEx(lpDirectoryName: PChar;  
    var lpFreeBytesAvailableToCaller, lpTotalNumberOfBytes;  
    lpTotalNumberOfFreeBytes: PLargeInteger): BOOL; stdcall;
```

Параметр `pDirectoryName` – ім'я каталога на ЕНІ, від якого необхідно одержати інформацію. Це не обов'язково буде кореневий каталог. Якщо:

lpDirectoryName=nil,

то отримаємо інформацію про поточний ЕНІ.

Параметр lpFreeBytesAvailableToCaller – це вільне місце (в байтах) на ЕНІ, доступне додатку.

Параметр lpTotalNumberOfBytes – ємність ЕНІ.

Параметр lpTotalNumberOfFreeBytes – покажчик на вільне число байтів на ЕНІ.

Інформацію про файлову систему, яка розташована на ЕНІ, будемо одержувати за допомогою функції GetVolumeInformation, об'явивши її наступним чином:

```
function GetVolumeInformation(lpRootPathName: PChar;  
    lpVolumeNameBuffer: PChar; nVolumeNameSize: DWORD;  
    lpVolumeSerialNumber: PDWORD;  
    var lpMaximumComponentLength, lpFileSystemFlags: DWORD;  
    lpFileSystemNameBuffer: PChar;  
    nFileSystemNameSize: DWORD): BOOL; stdcall;
```

Параметр lpRootPathName – це кореневий каталог ЕНІ.

Параметр lpVolumeNameBuffer – буфер розміру nVolumeNameSize, в який повертається ім'я ЕНІ.

В параметр lpVolumeSerialNumber: повертається серійний номер ЕНІ (але це не заводський номер вінчестера. Це чисто службовий номер логічного диска).

В параметр lpMaximumComponentLength повертається допустиме число символів в елементах імен файлів, тобто – число символів, які можуть розташовуватись між обраними слешами. Наприклад, якщо підтримуються довгі імена, то

lpMaximumComponentLength = 25.

В параметр lpFileSystemFlags повертаються прапорці, що характеризують файлову систему на ЕНІ. Параметром в системі буде сполученням наступних прапорців:

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

- FS_CASE_IS_PRESERVED - система зберігає регістр, в якому задані імена файлів, розміщених на ЕНІ.

- FS_CASE_SENSITIVE - система підтримує чутливість імен до регістру.

- FS_UNICODE_STORED_ON_DISK - система підтримує символи Unicode в файлах, розміщених на ЕНІ.

- FS_PERSISTENT_ACLS - система підтримує ACL.

- FS_FILE_COMPRESSION - система підтримує стисненні файли.

- FS_VOL_IS_COMPRESSED - даний том стиснений.

В буфер lpFileSystemNameBuffer розміру nFileSystemNameSize повертається ім'я файлової системи, наприклад, «FAT» чи «NTFS».

Означимо шляхи реалізації алгоритму одержання інформації про файлову систему ЕНІ, використавши вищезначені функції.

Першим кроком буде занесення на форму інформації про ЕНІ ПК в вікно RichEdit1. Це виконується натисненням кнопки Button1. Оброблення натиснення реалізуємо за функцією CreateFile(lpFileName. Означимо синтаксис цієї функції:

```
function CreateFile(lpFileName: PChar;  
dwDesiredAccess, dwShareMode: DWORD;  
lpSecurityAttributes: PSecurityAttributes;  
dwCreationDisposition,  
dwFlagsAndAttributes: DWORD;  
hTemplateFile: THandle): THandle;
```

При доступі до портів параметр lpFileName утримує ім'я відкриваємого порту: COM/LPT/USB, а параметр dwDesiredAccess визначає прапорці доступу до нього. При цьому він може утримувати комбінацію наступних прапорців:

- 0 - дозвіл опитування атрибутів порту;

- GENERIC_READ - дозвіл на зчитування;

- GENERIC_WRITE - дозвіл на запис.

Комбінація прапорців за допомогою операції `or` дозволить відкрити порт одночасно і для запису, і для читання.

Параметр dwShareMode визначає режим сумісного доступу. Він завжди в системі буде мати значення «0», оскільки до портів сумісний доступ неможливий. По цій же причині повинні бути рівними nil параметри lpSecurityAttributes

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

(визначення успадкування породженими процесами) та hTemplateFile (шаблон і атрибути створюваного файлу).

Параметр dwCreationDistribution, який визначає дії зі створюваним файлом, буде для портів завжди мати значення OPEN_EXISTING// - відкриття існуючого порту. Параметр dwFlage визначає режим роботи:

- 0 – синхронний;
- FILE_FLAG_OVERLAPPED – асинхронний.

Асинхронному параметру привласнимо значення «0», оскільки в системі передбачається читання/запис в синхронному режимі з індикацією. В випадку успішного виконання, функція CreateFile повертає дескриптор порту. Успішність виконання будемо перевіряти, використавши функцію GetLastError, а після завершення роботи порт буде закритий функцією CloseHandle.

Функції API Windows повертають результат, який дозволяє зрозуміти, виникла помилка при її виконанні, чи ні. Це може бути одне із значень: false – помилка; NuLL та інші. При цьому ОС виробляє код помилки у вигляді 32-розрядного числа.

Для того, щоб одержати код помилки, використаємо функцію GetLastError, яка забезпечить повернення коду останньої помилки даного потоку. Ці коди індивідуальні для кожного потоку і інші роботи їх не змінюють. Аргументом при виклику цієї функції задається код помилки, який буде утримувати 1 в 29-му біті. Цей біт вказує, що код задається додатком. Оскільки в системних кодах 29-й біт дорівнює 0, ми використаємо 1. Це буде гарантувати, що наш код не перекриється з системним кодом.

Функція GetLastError буде викликатись зразу після повернення тої бібліотечної функції, помилку якої будемо визначати. Це пов'язано з тим, що деякі функції викликають при успішному завершенні SetLastError (0), що знищує код помилки.

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

Таким чином, розробці в процесі виконання ВКРБ підлягала досить складна (в сенсі – реалізації) система, тому в процесі проектування ПЗ використані функції Windows API – Win 64 та DLL-бібліотек.

DLL-бібліотеки забезпечить пряме звертання програм до необхідних процедур підпрограм чи функцій. Ми використаємо при розробці ПЗ наступні бібліотеки Win 64 API ядра ОС Windows: kernel 64, User 64, Advapi 64.

Окрім цього, в процесі роботи системи будуть використовуватись інтерфейси системних викликів ОС Windows, бібліотека драйверів пристроїв (на які необхідно створити образи) та рівні: NativeAPI (Ntdll.dll), NativeAPI ядра (ntoskrnl.exe), рівень абстракцій від обладнання (hal.dll). Це дозволяє організувати взаємодію ПЗ системи з пристроями для реалізації основної задачі – створення їх образів.

Реалізація зазначеного вище алгоритму дозволила розробити ПЗ, яке забезпечує створення/збереження/копіювання образів ЕНІ.

При розробці ПЗ виконана також одна із задач, що стояла перед розробником – реалізована максимальна ступінь автоматизації виконання робочих функцій.

Створений автоматичний режим інформації дозволяє втручання користувача тільки для ухвалення таких рішень і проведення таких дій, які не були заздалегідь визначені алгоритмами: інтерактивний режим роботи системи, реагування користувача на позаштатні ситуації в роботі ЗОТ чи системи.

Користувач не може заздалегідь знати стан ПК в кожен момент часу і, відповідно, своєчасно прийняти заходи по стабілізації його роботи, тому розроблене ПЗ автоматично виконує наступні функції: розподіл часу і зовнішніх пристроїв між задачами, виділення пристрою введення/виводу інформації/даних. Передбачено і контроль граничних умов: перевірка доступності ресурсу для отримання і збереження інформації.

Для забезпечення високої надійності функціонування системи проводиться обробка помилок: виключені проблеми, відомі як витік пам'яті; гарантовано

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

звільняються ресурси ПК після їх використання розробленим ПЗ; гарантований рівень надійності та ефективності при обробці помилкових станів системи.

Розглянемо роботу основних алгоритмів системи.

Головна програма клієнта, по суті - диспетчер системи. Вона виконує керування модулями системи. Для проведення операцій низького рівня вона використовує API - функції ОС Windows та програмні інтерфейси операційної системи. Це дозволяє зробити доступними зовнішнім модулям підключені до головної програми модулі, оголошені типи, класи, змінні, функції і процедури, які в процесі роботи звертаються до головної програми.

Розглянемо принцип побудови та роботу головної програми, блок – схема алгоритму якої представлена на рисунку 4.1.

Наводимо алгоритм роботи головної програми:

- Крок 1: початок роботи головної програми. Перехід на крок 2
- Крок 2: якщо надійшла команда з командного рядка – перехід на крок 3, якщо ні – перехід на крок 7.
- Крок 3: встановлення прапорців відповідно до опції командного рядка. Перехід на крок 4.
- Крок 4: якщо прапорці встановлені відповідно до опції успішно – перехід на крок 5. Якщо ні - виведення повідомлення про помилку і перехід на крок 22.
- Крок 5: обробка параметрів командного рядка. Перехід на крок 6;
- Крок 6: якщо обробка параметрів пройшла успішно – перехід на крок 7. Якщо ні – виведення повідомлення про помилку і перехід на крок 22.
- Крок 7: виконання команди з командного рядка. Перехід на крок 21.
- Крок 8: введення пароля користувачем. Перехід на крок 9.
- Крок 9: якщо пароль вірний – перехід на крок 10, якщо ні – виведення повідомлення про помилку і перехід на крок 21.
- Крок 10: створення індексованого списку TStringList. Перехід на крок 11.
- Крок 11: пошук дисководів для оптичних дисків. Перехід на крок 12.

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

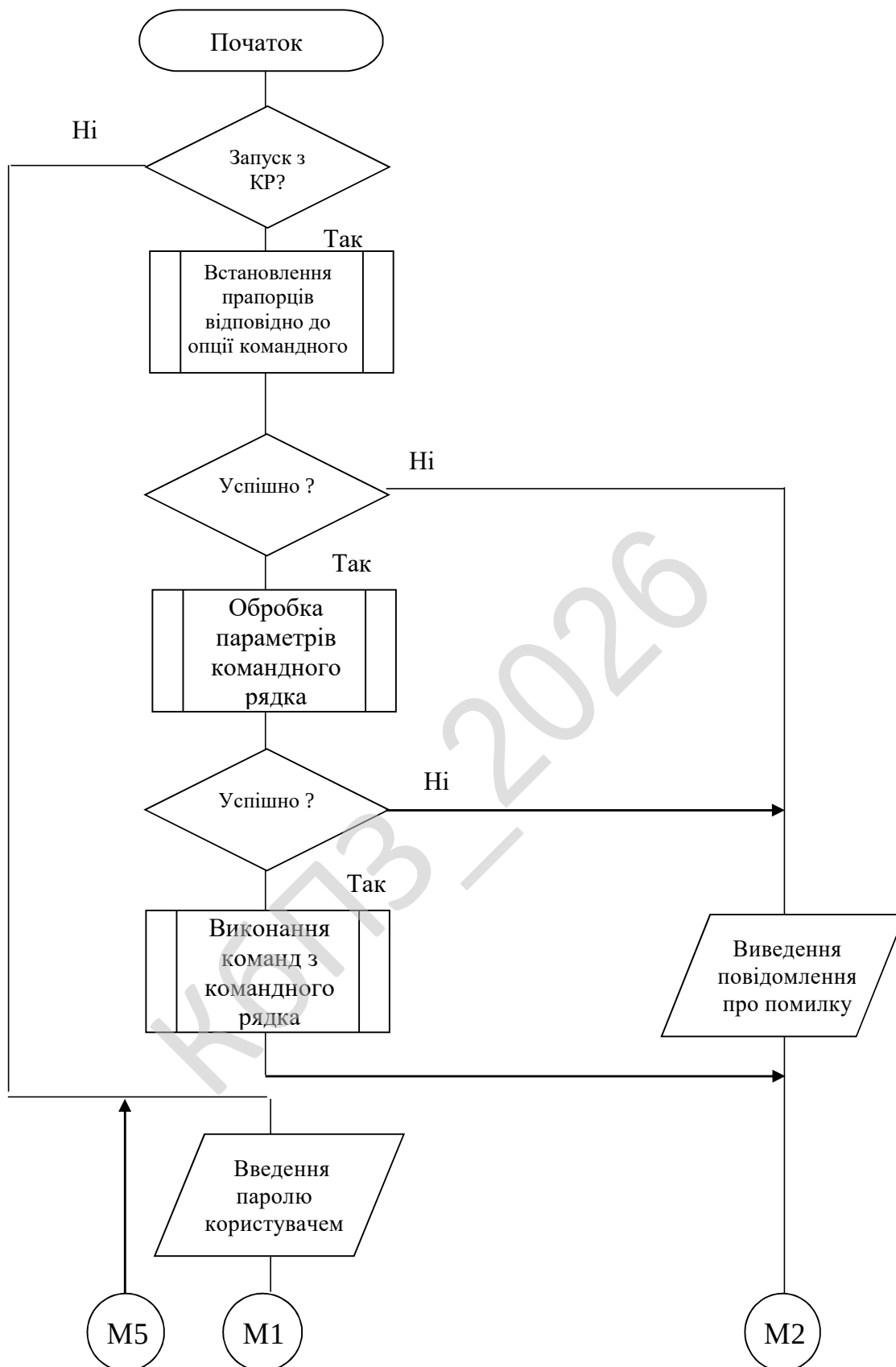


Рисунок 4.1 - Блок-схема алгоритму роботи головної програми

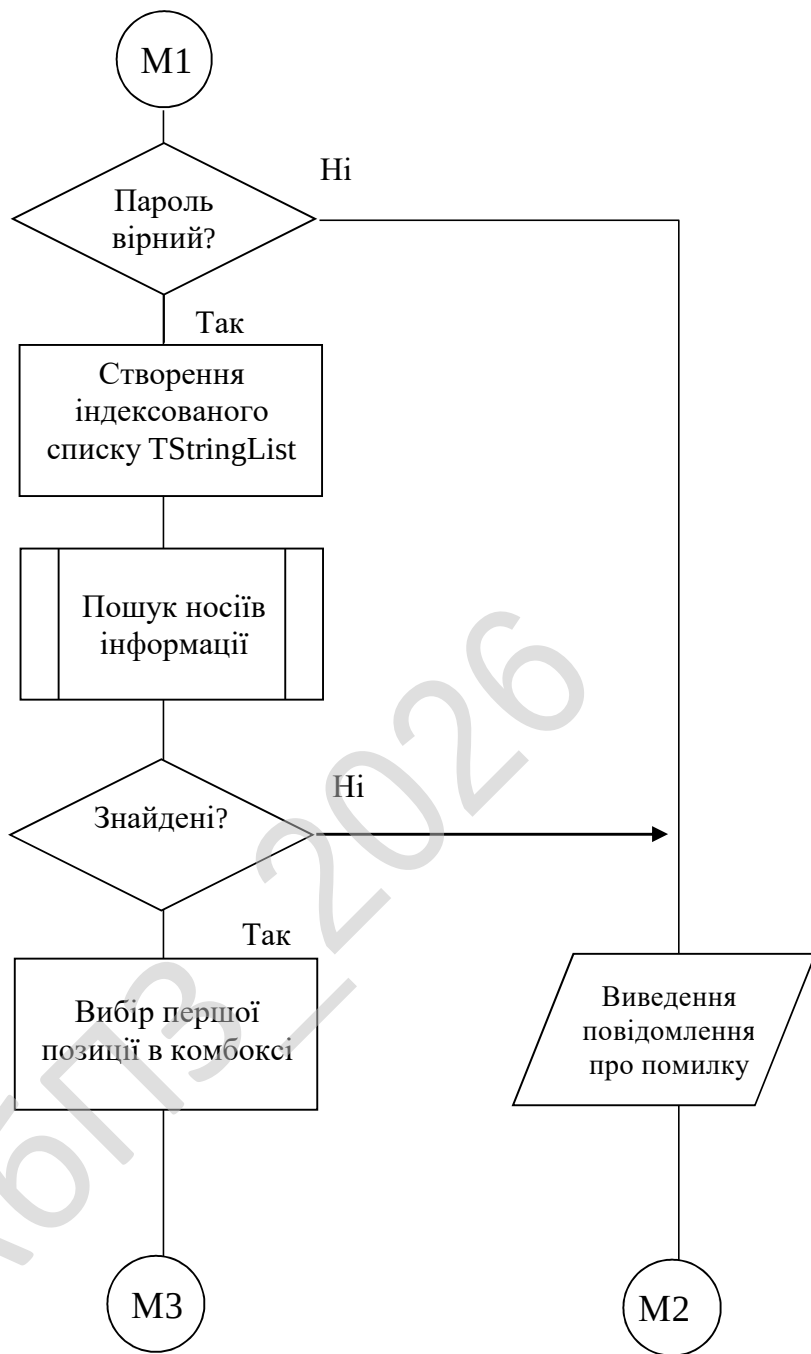


Рисунок 4.1, аркуш 2

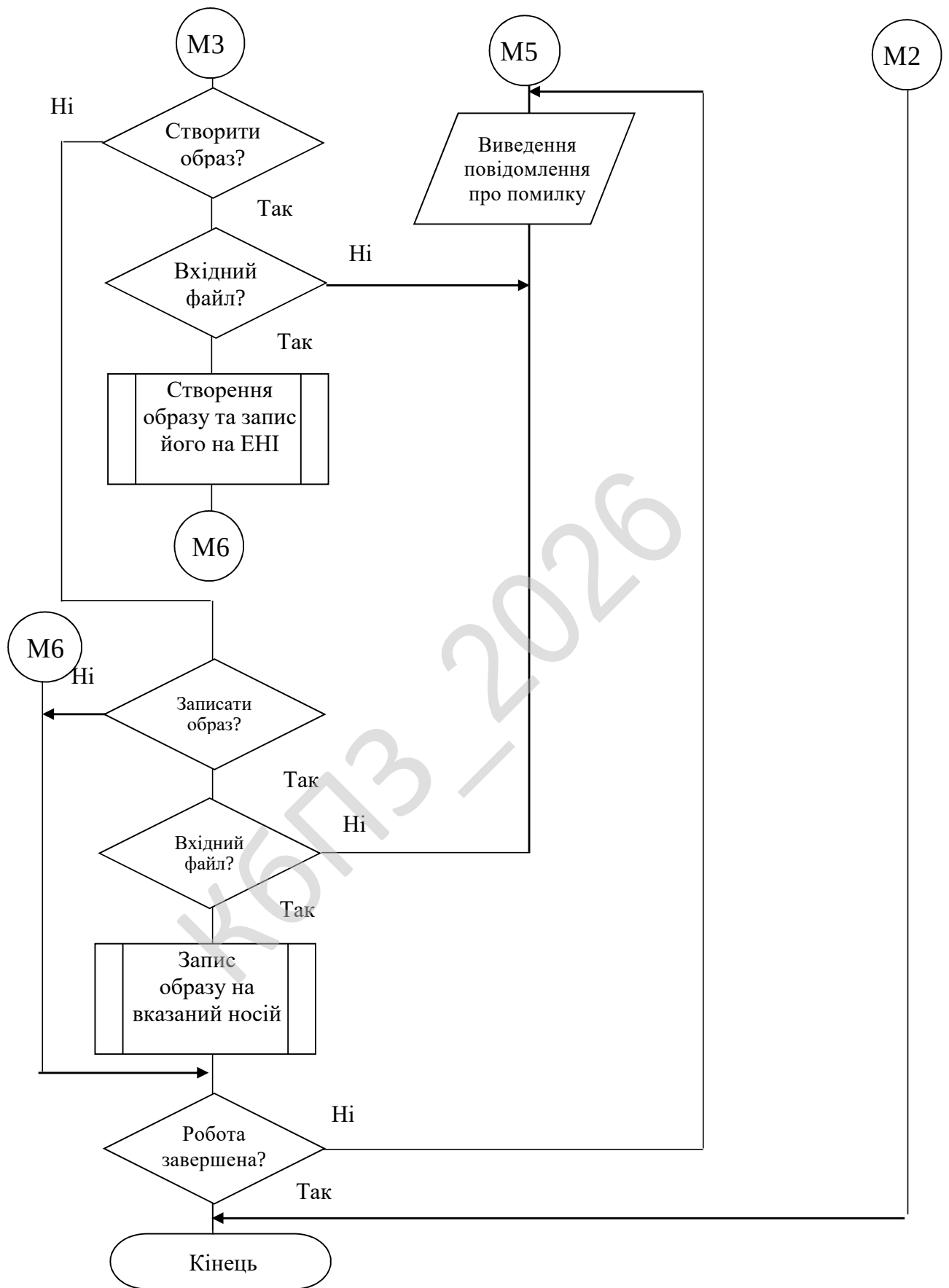


Рисунок 4.1, аркуш 3

- Крок 12: якщо дисководи знайдені – перехід на крок 13, якщо ні – виведення повідомлення про помилку і перехід на крок 21.
- Крок 13: вибір першої позиції в комбобоксі. Перехід на крок 14.
- Крок 14: якщо користувач бажає створити образ – перехід на крок 15, якщо ні – перехід на крок 17.
- Крок 15: якщо вхідний файл обрано – перехід на крок 16, якщо ні - виведення повідомлення про помилку і повернення на крок 8;
- Крок 16: створення образу файла і запис його на диск вінчестера. Перехід на крок 20.
- Крок 17: якщо користувач бажає записати образ – (раніше створений, але не записаний) – перехід на крок 18, якщо ні – перехід на крок 20.
- Крок 18: якщо вихідний образ обрано (з вінчестера) – перехід на крок 19, якщо ні – повернення на крок 8.
- Крок 19: запис образу на вказаний носій. Перехід на крок 20.
- Крок 20: якщо користувач бажає завершити роботу – перехід на крок 21, якщо ні – повернення на крок 8.
- Крок 21: кінець роботи головної програми.

Процедура читання інформації з фізичного сектору ЕНІ забезпечує в системі виконання функції посекторового читання пристрою, що по суті є створенням його образу. Наводимо блок-схему роботи вищезначеного алгоритму на рисунку 4.2.

Алгоритм роботи процедури читання інформації з фізичного сектору має наступний вигляд:

- Крок 1: початок роботи процедури. Перехід на крок 2.
- Крок 2: виведення інформації в рядок статусу. Перехід на крок 3.
- Крок 3: перевірка наявності хендла пристрою. Якщо хендл пристрою є – перехід на крок 4, якщо ні – виведення повідомлення про помилку і перехід на крок 15.

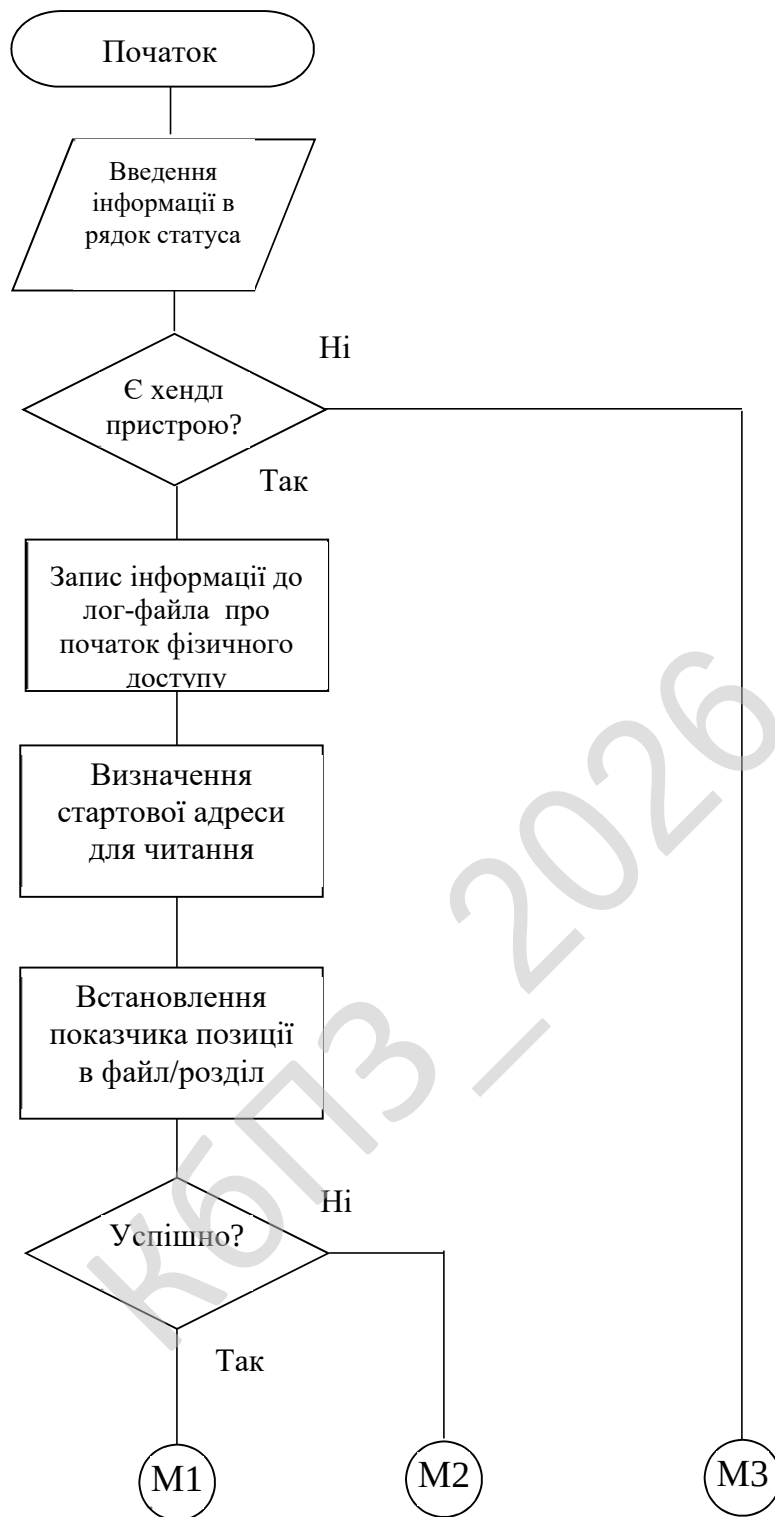


Рисунок 4.2 – Блок-схема алгоритму роботи процедури читання інформації з фізичного сектору

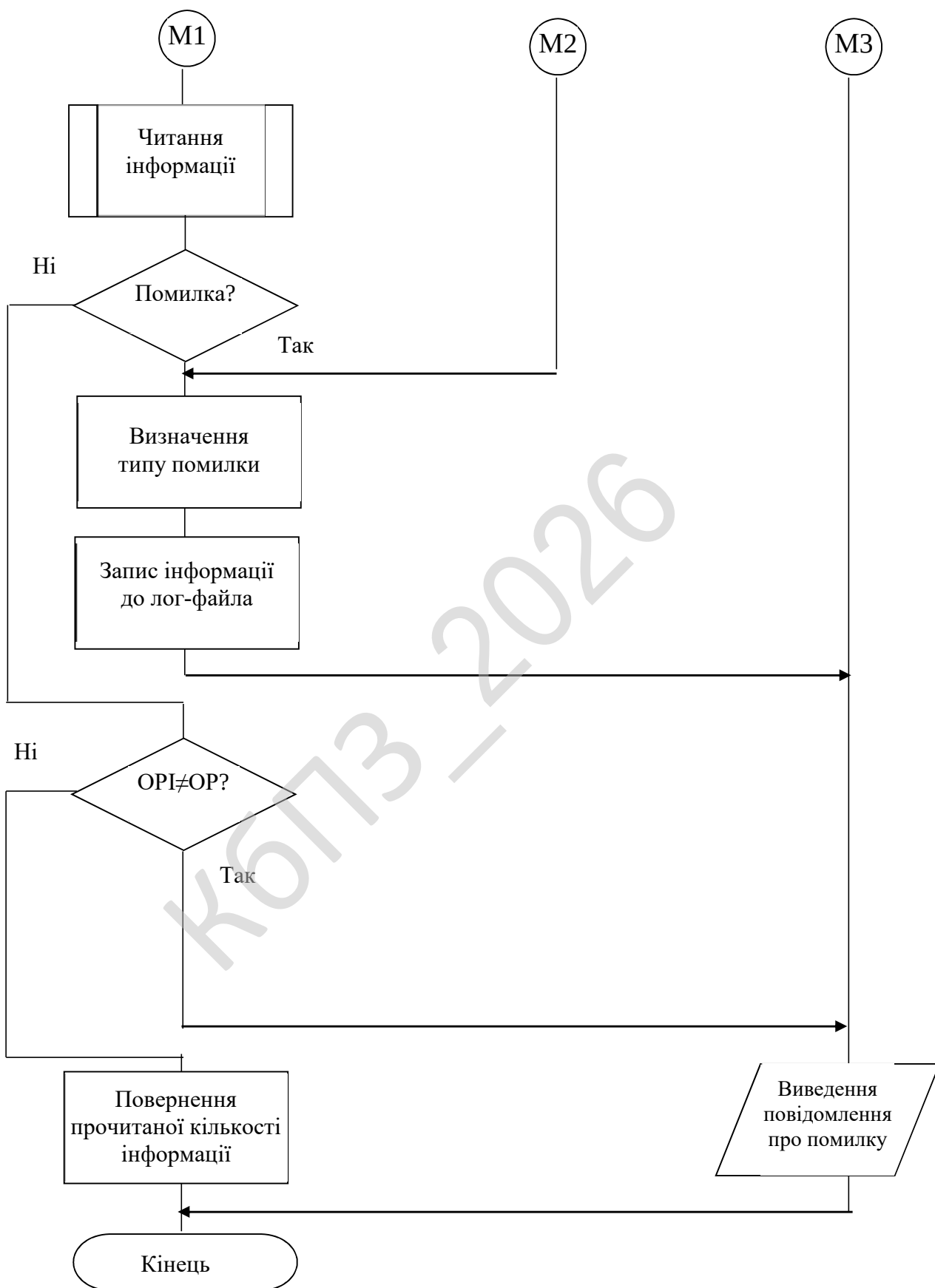


Рисунок 4.2, аркуш 2

- Крок 4: запис інформації до лог-файлу про початок фізичного доступу.

Перехід на крок 5.

- Крок 5: визначення стартової адреси розділу/файлу для читання. Перехід на крок 6.

- Крок 6: встановлення покажчика позиції в розділі чи файлі в конкретну позицію. Перехід на крок 7.

- Крок 7: якщо покажчик встановлений в коректну позицію успішно, можна переходити до кроку 8, якщо ні – перехід на крок 10.

- Крок 8: читання інформації з фізичного сектора розділу/файла. Перехід на крок 9.

- Крок 9: якщо спроба читання пройшла успішно – перехід на крок 12, якщо ні – перехід на крок 10.

- Крок 10: визначення типу помилки. Перехід на крок 11.

- Крок 11: запис інформації до лог-файлу. Перехід на крок 15.

- Крок 12: розрахунок обсягу очікуваного розміру інформації та реально отриманого обсягу інформації. Якщо одержані значення відповідні - перехід на крок 14, якщо ні – перехід на крок 13.

- Крок 13: виведення повідомлення про помилку. Перехід на крок 15.

- Крок 14: повернення прочитаної кількості інформації. Перехід на крок 15.

- Крок 15: кінець роботи процедури.

Після того, як був створений образ ЕНІ, його необхідно зберегти на жорсткий диск. Для цього в процесі виконання ВКРБ був розроблений і програмно реалізований алгоритм процедури запису фізичного сектора ЕНІ, блок-схема якого наведена на рисунку 4.3.

Алгоритм роботи процедури запису фізичного сектора ЕНІ має наступний вигляд:

- Крок 1: початок роботи процедури. Перехід на крок 2.

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

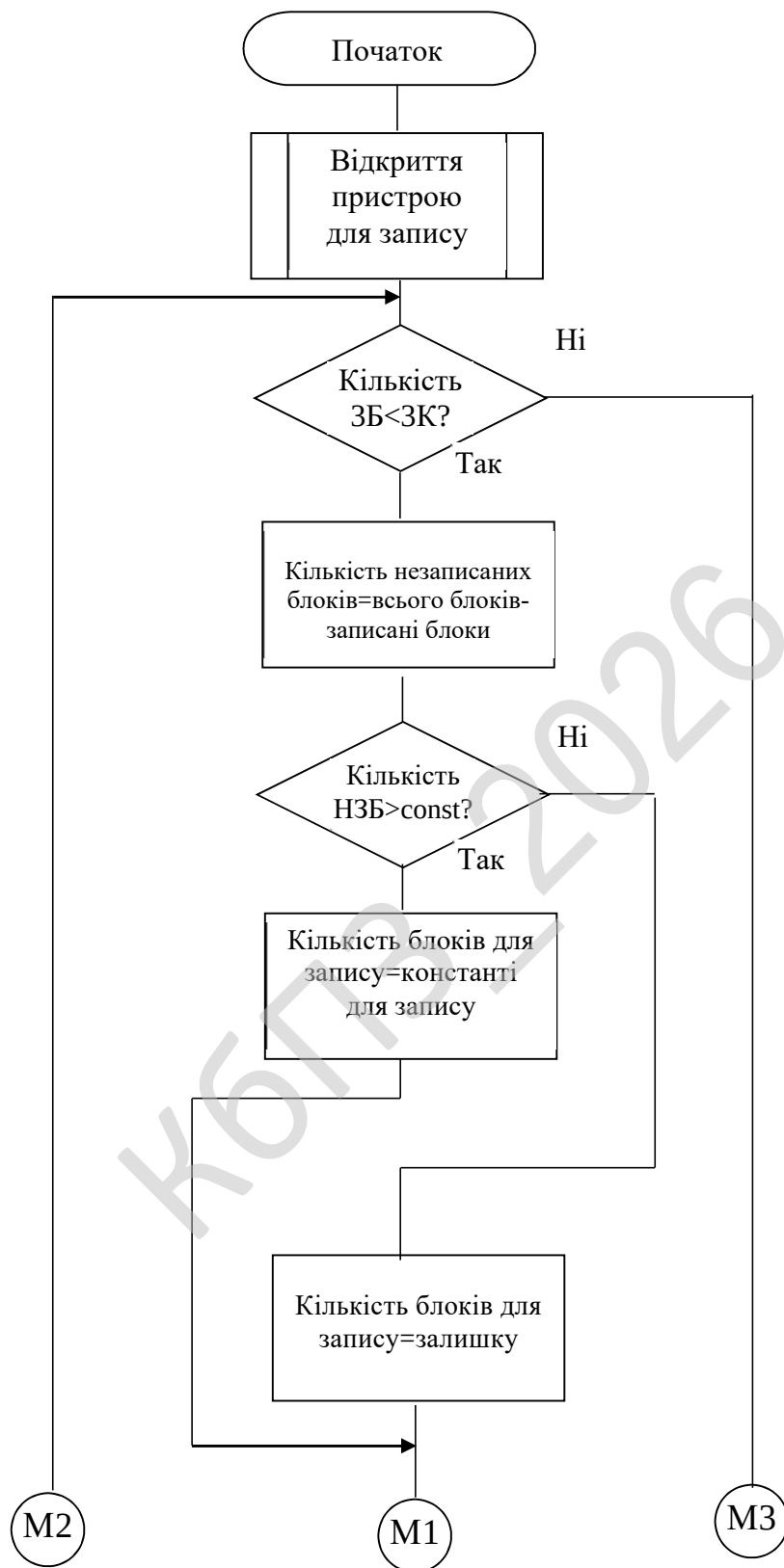


Рисунок 4.3 – Блок-схема алгоритму роботи процедури запису фізичного сектора ЕНІ

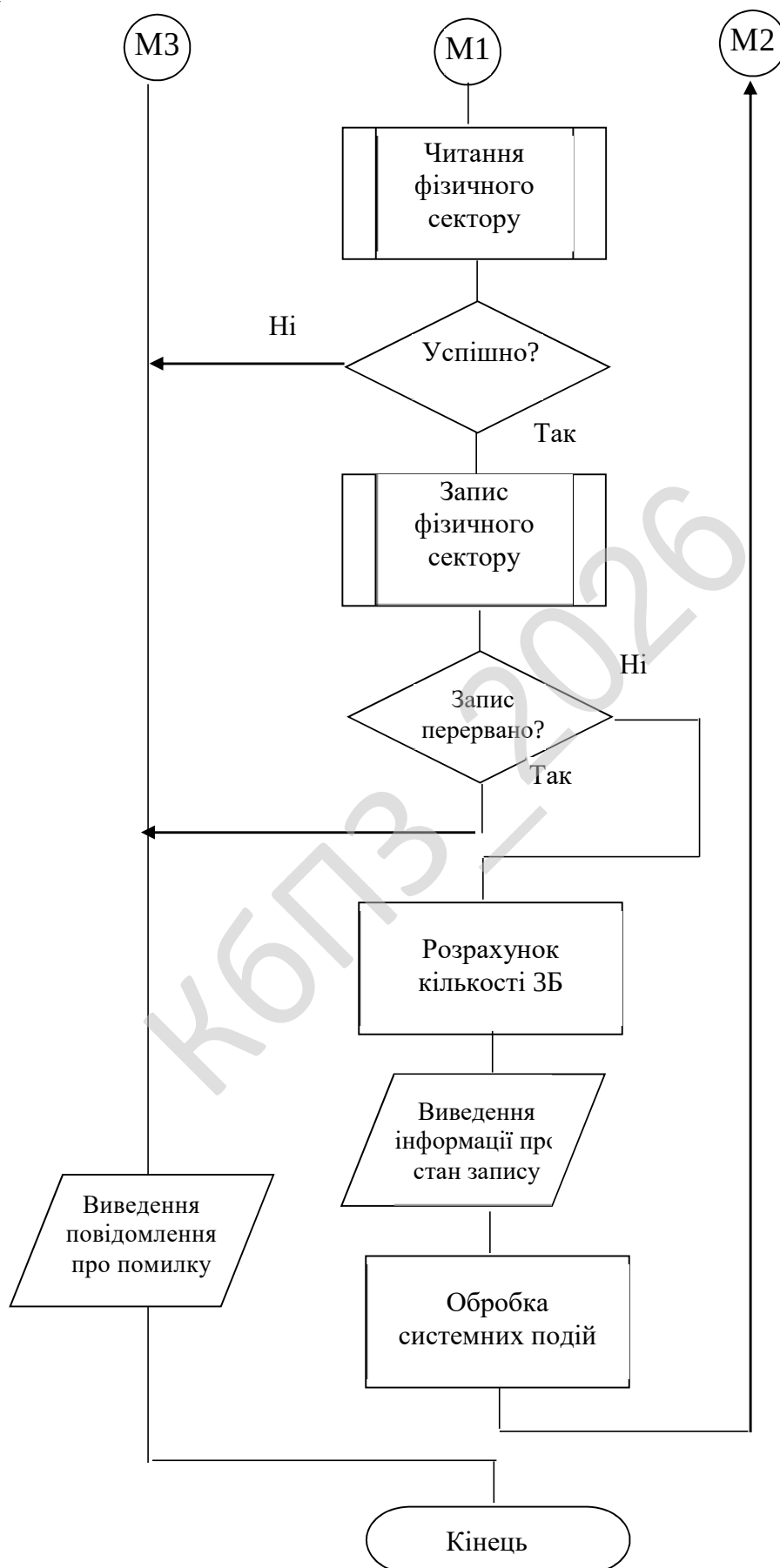


Рисунок 4.3, аркуш 2

- Крок 2: відкриття пристрою (МНІ для запису). Перехід на крок 3.
- Крок 3: якщо кількість записаних на пристрої блоків менша загальної кількості – перехід на крок 4, якщо ні – на крок 15.
- Крок 4: виконання розрахунку кількості незаписаних блоків, яка дорівнює: всього блоків – записані блоки. Перехід на крок 5.
- Крок 5: якщо кількість незаписаних блоків більша означеної константи запису блоків – перехід на крок 6, якщо ні – на крок 7.
- Крок 6: кількості блоків для запису привласнюємо значення константи для запису. Перехід на крок 8.
- Крок 7: кількості блоків для запису привласнюємо значення залишку. Перехід на крок 8.
- Крок 8: читання файла/розділу розміром в кількості блоків для запису. Перехід на крок 9.
- Крок 9: якщо процедура пройшла успішно – перехід на крок 10, якщо ні – на крок 15.
- Крок 10: запис фізичного сектора розміром, заданим кількістю блоків для запису. Перехід на крок 11.
- Крок 11: якщо запис примусово перервано користувачем – перехід на крок 15, якщо ні – на крок 12.
- Крок 12: розрахунок кількості записаних блоків. Перехід на крок 13.
- Крок 13: візуалізація інформації про стан запису. Перехід на крок 14.
- Крок 14: обробка системних подій. Перехід на крок 15.
- Крок 15: кінець роботи процедури.

В якості ілюстрації наводимо фрагмент програмного коду процедури запису фізичного сектора ЕНІ:

```
if Device.Open then
    try
        try
            // запис
            while WrittenBlocks < Blocks do
```

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

```

begin
    // кількість блоків, що залишилась
    BlocksRemaining := Blocks - WrittenBlocks;
    if BlocksRemaining > BlocksCount then
    begin
        BlockCount := BlocksCount;
    end
    else
    begin
        BlockCount := BlocksRemaining;
    end;
    // читати файл
    ReadFile2(h1, PChar(Buffer), tGem.BytesPerSector *
BlockCount, Read, nil);
    if Read = 0 then break;
    try
        // запис фізичного сектора
        Device.WritePhysicalSector(WrittenBlocks,
BlockCount, PChar(Buffer));
    except
        StopRead := True ;
        break ;
    end ;
    // кількість записаних блоків
    WrittenBlocks := WrittenBlocks + BlockCount;
    // інформація для статус - бару
    StatusBar1.Panels[0].Text := IntToStr((WrittenBlocks * 100) div Blocks) +
'%';

    // дозволити системі виконати чергу повідомлень
    Application.ProcessMessages;
end;
except
    on E : Exception do
    begin
        MessageDlg(E.Message, mtError, [mbOK], 0);
        HadError := True;
    end;
end;
finally
    // закрити пристрій
    Device.Close;
    // звільнити об'єкт
    Device.Free;
end
else
begin
    // отримати лінк на помилку
    Error := GetLastError;
    // показати текст помилки
    MessageDlg(Format(ErrorOpeningDisk, [GetLastError]) + #10 +
SysErrorMessage(Error) , mtError, [mbOK], 0);
    // встановити прапорець помилки
    HadError := True;
end;
end;

```

Програмне забезпечення, розроблене згідно ТЗ, повністю працездатне. Це показала виконана в рамках ТЗ дослідна експлуатація системи. Тому можна зробити висновок, що запланований обсяг робіт виконаний на 100 %; розроблена

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

достатньо потужна система створення та збереження образів ЕНІ, що є на сьогоднішній день достатньо актуальним питанням.

Для більш детальної ілюстрації роботи системи розроблено і виконано 7 аркушів формату А4 графічного матеріалу: структурна схема системи, функціональна схема системи, діаграма процесів, блок-схеми алгоритмів головної програми і програм, які найефективніше характеризують розроблене ПЗ.

Подальше вдосконалення системи можливе в двох напрямках:

- введення в систему нових модулів з метою розширення її функцій і надання нових послуг користувачу;
- введення в систему елементів штучного інтелекту для формування в автоматичному режимі початкових умов і вимог до роботи системи.

Доопрацювання системи не буде трудомістким завданням, оскільки розроблене ПЗ має модульну структуру побудови, внаслідок чого система одержала високу ступінь мобільності і адаптивності.

4.2 Захист розробленого програмного забезпечення

На сьогоднішній день реальний захист авторських прав на комп'ютерні програми можна забезпечити лише встановленням засобів програмного захисту. Задача розробника ПЗ полягає в тому, щоб користувач, встановивши піратську копію ПЗ на комп'ютер, не зміг би використовувати її внаслідок того, що спрацюють засоби програмного захисту від нелегального копіювання. Це задача програмістів, і вона повинна бути реалізована на етапі розробки ПЗ або на початку промислової експлуатації системи на підприємстві (у організації).

При розробці програми захисту інформації було вирішено організувати санкціонований доступ до роботи з системою. Перш, ніж отримати дозвіл на роботу з системою, користувач-абонент повинен на вимогу системи назвати свій пароль. Пароль встановлюється розробником системи або системним адміністратором організації-користувача. Еталон пароля розміщується в базі даних

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

в спеціальному файлі. При введенні пароля власне користувачем, проводиться його порівняння з еталоном. В результаті цієї процедури можливі два випадки:

- пароль співпадає з еталоном;
- пароль не співпадає з еталоном.

Наслідок: система в обох випадках реагує однаково – повторно пропонує ввести пароль для підтвердження права доступу до системи.

У першому випадку – це ще один ступінь захисту системи, тобто - користувач знає, що код доступу він повинен ввести n-у кількість разів (кількість введень встановлюється системним адміністратором) і якщо код доступу вірний – отримує право роботи з системою.

Якщо пароль не співпадає з еталоном, то після повторного неправильного його введення, версія ПЗ, що знаходиться на даному комп'ютері, буде знищена. Такі жорсткі дії були введені для захисту програмного забезпечення від “злому” встановленого захисту.

Цього ступеню захисту розробленого ПЗ на даному етапі цілком достатньо. Складніший захист може бути встановленим у разі виникнення попиту на розроблений програмний продукт.

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		80

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Відповідно до технічного завдання в процесі роботи над випускною кваліфікаційною роботою за першим (бакалаврським) рівнем вищої освіти було розроблене програмне забезпечення автоматизованої системи створення та збереження образів електронних носіїв інформації на основі використання функцій DeviceIoControl ОС Windows.

Методика впровадження розробленої системи передбачає комплексний підхід: інвентаризацію активів, розробку політик безпеки та резервування, розгортання технічних засобів, створення «золотого образу» (еталону образу диска), тестування відновлення та регулярний аудит.

Важливим є автоматизація та регулярне тестування відновлення. Це забезпечить ефективну роботу розробленої системи по безпосередньому призначенню та швидке відновлення працездатності ПЗ та даних в разі їхнього пошкодження. Наводимо перелік принципових підходів до налаштування розробленої системи на ПК Замовника. Основні етапи методики впровадження:

1. Аналіз та планування:

- Визначення критично важливих даних та носіїв, вибір типу образів (повний, диференціальний) та місця зберігання (локально, мережа, хмара).

- Встановлення цільового часу відновлення (RTO) та цільової точки відновлення (RPO).

- Розробка регламента резервного копіювання.

2. Підготовка технічного середовища:

- Встановлення та налаштування ПЗ для створення образів, забезпечення захищеного сховища.

- Підготовка сховищ (мережеві диски, NAS, хмарні сервіси).

3. Технічне впровадження:

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

- Налаштування автоматизованого створення образів за розкладом для запобігання втраті даних.

- Налаштування інкрементальних або диференціальних образів (для оновлень). Інкрементальні та диференціальні образи (зображення) - це методи обробки даних та формування резервних копій, які базуються на фіксації змін. Диференціальні образи зберігають зміни відносно останньої повної копії, що пришвидшує відновлення, тоді як інкрементальні фіксують лише зміни з моменту останнього будь-якого образу (що швидше створюється і займає менше місця).

- Створення базового образу («золотий образ»): налаштування еталонної ОС з необхідним ПЗ та драйверами, створення її образу.

- Шифрування образів для забезпечення конфіденційності.

4. Зберігання (Правило 3-2-1): мати не менше три копій даних. Зберігати на двох різних типах носіїв. Одна копія повинна знаходитися поза межами офісу (off-site або хмара).

5. Тестування та підтримка:

- Регулярна перевірка цілісності образів шляхом тестового відновлення даних для перевірки працездатності системи.

- Регулярна перевірка можливості розгортання образів на «чисте» залізо.

6. Документування та навчання: розробка інструкцій для персоналу щодо процедур створення та відновлення образів.

Ця методика забезпечує мінімізацію ризиків втрати даних при збоях обладнання, кібератаках або людських помилках. Ключовим фактором є забезпечення цілісності (хешування) та безпеки (шифрування) збережених образів.

Образи електронних носіїв (файли-копії вмісту дисків) зберігаються на захищених серверах, у системах електронного документообігу, на хмарних сховищах, зовнішніх жорстких дисках (NAS) або в спеціалізованих електронних архівах. Основна вимога - забезпечення цілісності, доступності та можливості перевірки електронного підпису протягом щонайменше 1095 днів.

Поширені місця зберігання образів електронних носіїв інформації:

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

- Хмарне сховище (Cloud Backup): надійний метод, який захищає дані від фізичного знищення пристрою (Dropbox, Google Drive, OneDrive).

- Зовнішні носії: зовнішні жорсткі диски (HDD/SSD) та USB-флешки - швидкий локальний спосіб.

- NAS-системи (Network Attached Storage): мережеві сховища для автоматичного резервного копіювання даних з ПК.

- Електронні архіви: Спеціально організовані місця зберігання для забезпечення тривалого зберігання та юридичної значущості.

Рекомендуємо використати стратегію «Правило 3-2-1», яка переважно вживається, оскільки гарантує максимальну безпеку:

- Мати не менше 3 копій даних.

- Зберігати їх на 2 різних типах носіїв.

- 1 копію тримати поза межами робочого місця (наприклад - у хмарі).

Інтеграція системи захисту в існуючу інфраструктуру Замовника передбачає сумісність із іншими платформами, наявність API для розширення функціональності, централізоване управління та масштабованість для адаптації до змін у мережі. Відсутність належної інтеграції може призвести до прогалин у безпеці та конфліктів між компонентами.

Інтеграція також вимагає тісної взаємодії з системами управління ідентифікацією та доступом для забезпечення чіткої сегментації прав доступу. Це дозволяє мінімізувати ризики, пов'язані з людським фактором, і підвищити прозорість доступу до критичних активів.

Експлуатацію розробленої системи буде здійснювати персонал підрозділів і відповідних служб організації-замовника. Обслуговування та розвиток системи має здійснювати спеціалізований підрозділ, який необхідно створити в процесі впровадження системи в промислову експлуатацію. Цей підрозділ може бути цілком самостійним (або входити до складу служби, що безпосередньо буде експлуатувати розроблену систему) і має забезпечити виконання наступних задач:

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83

- технологічне супроводження процесу виконання безпосередніх функціональних задач системою;
- системне забезпечення інформаційно-обчислювальних процесів;
- розробку сервісних програмних компонентів та доробку, в разі необхідності, програмного забезпечення системи з цілю його модифікації;
- супроводження програмних компонентів;
- обслуговування технічних засобів;
- організацію робіт по розвитку системи.

В якості ілюстрації роботи системи по безпосередньому призначенню наводимо скриншоти ПЗ (рисунок 5.1, рисунок 5.2, рисунок 5.3, рисунок 5.4, рисунок 5.5).

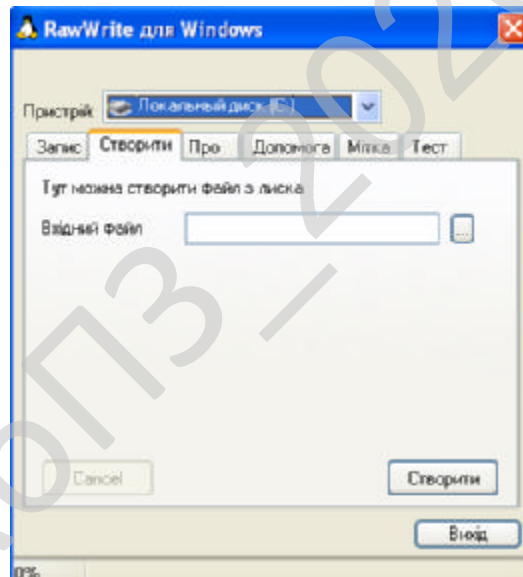


Рисунок 5.1 – Скриншот вікна створення образу диску

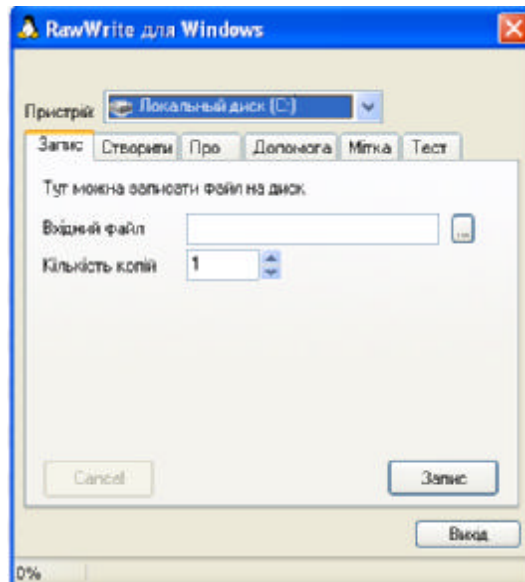


Рисунок 5.2 – Скриншот вікна запису збереженого образу диску

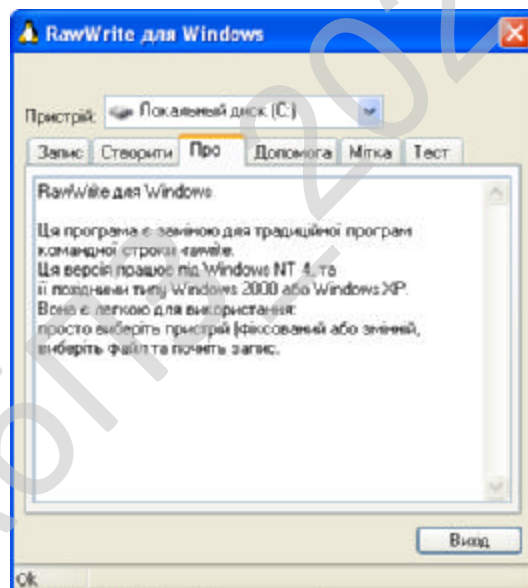


Рисунок 5.3 – Скриншот вікна інформації про програму

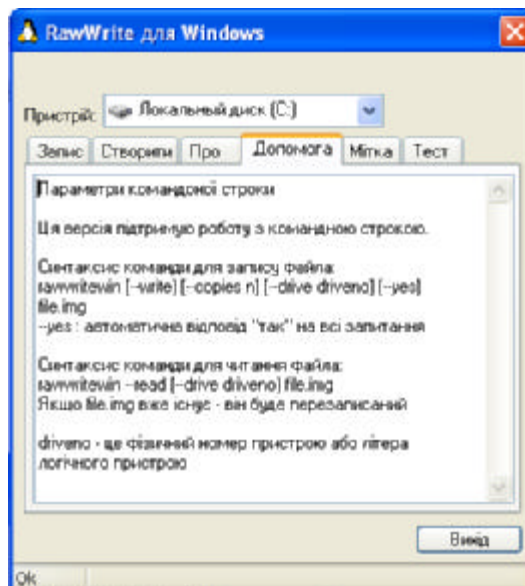


Рисунок 5.4 – Скриншот вікна синтаксису командного рядка

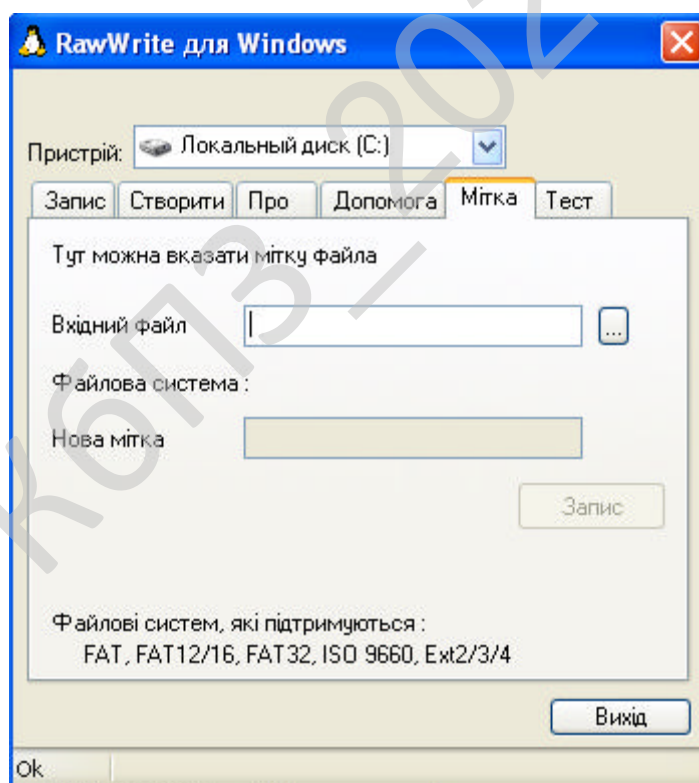


Рисунок 5.5 – Скриншот вікна зміни мітки файла

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, розроблене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначене для автоматизованої системи створення та збереження образів електронних носіїв інформації. Образ диска - це файл, що містить точну побайтову копію всього вмісту та структури фізичного носія (CD, DV, жорсткого диска, флешки), включаючи файлову систему, файли та завантажувальні записи.

Вітчизняні розробки програмних систем аналогічного класу та спрямування в межах України представлені не в достатній мірі, хоча на сьогодні це питання є надзвичайно актуальним.

Виконання завдання на розробку ПЗ системи полягало у вирішенні наступних задач:

- проведення огляду наявних на світовому ринку програмних продуктів систем - аналогів
- дослідження механізму створення та збереження образів електронних носіїв інформації на основі використання функцій DeviceIoControl ОС Windows.
- виконання на основі отриманих результатів досліджень програмної реалізації створення та збереження образів електронних носіїв інформації на основі використання функцій DeviceIoControl ОС Windows.

Розроблені алгоритми, дозволяють успішно вирішувати задачі створення та збереження інформації у вигляді образів ЕНІ (цифрових копій/образів дисків) з метою захисту даних від фізичного знищення, кібератак. Це дозволяє оперативно робити резервні копії, гарантує цілісність документів, файлової структури ЕНІ та забезпечує доступ до інформації незалежно від фізичного місцезнаходження.

Розроблена система є:

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		87

- апаратно невибагливою, що дає можливість працювати як на одному, окремо взятому ПК, так і на складній комбінації ПК локальних мереж, що працюють на віддаленому доступі один від одного;

- дозволить зберегти раніше зроблені капіталовкладення при переході на більш продуктивний варіант апаратної платформи;

- забезпечує можливість роботи в режимі реального часу;

- забезпечує обмін даними з іншими ПК КМ;

- надійно захищена від несанкціонованого доступу;

- має прийнятну вартість для мінімальної конфігурації комплексу технічних засобів і можливість використання обладнання, відповідного сучасним технічним досягненням.

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.

В якості середовища розробки ПЗ системи використане середовище швидкої розробки Delphi. Це одне з найбільш потужних візуально об'єктно-орієнтованих середовищ програмування, яке дозволяє на найсучаснішому рівні створювати розгалужені програмні комплекси, призначені для роботи в комп'ютерних мережах та в мережах Інтернет. Дана мова програмування дозволила ефективно обробити дані для системи та мінімізувати терміни розробки ПЗ. Розроблене ПЗ працює під керуванням ОС Windows.

Проведена в рамках ТЗ дослідна експлуатація системи показала, що вона цілком дієздатна та, завдяки наявності режиму HELP і україномовного графічного інтерфейсу користувача, дуже проста для сприйняття, хоча і вимагає від користувача певного рівня базової комп'ютерної грамотності.

Надаються необхідні рекомендації з установки розробленого ПЗ, проведення дослідної експлуатації та подальшої промислової експлуатації.

В цілому, розроблене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		88

завдання. Створене програмне забезпечення, маючи модульну структуру побудови, має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

К6ПЗ_2026

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		89

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Axelrod A. Complete Guide to Test Automation: Techniques, Practices, and Patterns for Building and Maintaining Effective Software Projects / A. Axelrod. – NY: Apress, 2018. – 588 p
2. Richards M. Fundamentals of Software Architecture: An Engineering Approach / M. Richards, N. Ford. – Sebastopol: O'Reilly Media, 2020. – 432 p.
3. Fowler M. UML Distilled: A Brief Guide to the Standard Object Modeling Language / M. Fowler. – Boston: Addison-Wesley Professional, 2018. – 191 p.
4. Rumbaugh J. UML 2.0. Object-Oriented Modeling and Design with UML / J. Rumbaugh, M. Blaha. – London: Pearson, 2012. – 496 p.
5. Larman C. Applying UML and Patterns: An Introduction to ObjectOriented Analysis and Design and Iterative Development / C. Larman. – London: Pearson, 2004. – 736 p.
6. . Aho A.V., Hopcroft J.E., Ullman J.D. Data Structures and Algorithms. – Pearson, 2001. – 620 p.
7. Beller M., Gousios G., Panichella A., Proksch S., Amann S., & Zaidman A. (2017). Developer testing in the ide: Patterns, beliefs, and behavior. IEEE Transactions on Software Engineering, 45(3), – 261-284 p.
8. IDS vs. IPS: Definitions, Comparisons & Why You Need Both: веб-сайт. URL: <https://www.okta.com/identity-101/ids-vs-ips/> (дата звернення: 17.11.2024).
9. Sichkar M., Pavlova L. A short survey of the capabilities of Next Generation firewalls. Computer Science and Cybersecurity. 2023. №1. – 28–33 p.
10. Knuth D. Art of Computer Programming, Vol. 2: Seminumerical Algorithms 3rd Edition, Kindle Edition. – Addison-Wesley Professional, 2019. – 672 p.
11. Knuth D. The Art of Computer Programming, Vol. 1: Fundamental Algorithms, 3rd Edition 3rd Edition. – Addison-Wesley Professional, 2019. – 672 p.

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

12. Cormen T.H., Leiserson C.E., Rivest R.L., Stein C. Introduction to Algorithms, 3rd Edition (The MIT Press) 3rd Edition – The MIT Press, 2019. – 1292 p.

13. «Про Стратегія інформаційної безпеки України» Указ Президента України: Стратегія від 28.12.2021 № 685/202. URL: <https://zakon.rada.gov.ua/laws/show/685/2021#Text>.

14. Новіков Ф.А. Навчально-методичний посібник з дисципліни Технологічні підходи до розробки програмного забезпечення/ Ф.А. Новіков. – СПб.: СПбГУ ІТМО, 2007. – 137 с.

15. Гагаріна Л. Технології розробки програмного забезпечення: навчальний посібник / Л. Гагаріна, Є. Кокорєва, Б. Віснадул. – М.: ВБ «ФОРУМ»: ІНФРА-М, 2008. – 400 с.

16. 10. Коробейнікова Т. І., Цар О. О. Аналіз сучасних відкритих систем виявлення та запобігання вторгнень. Грааль науки. 2023. № 27. – 317–325 с.

17. Жилін А. В., Шаповал О. М., Успенський О. А. Технології захисту інформації в інформаційнотелекомунікаційних системах: навч. посіб. Київ: КПІ ім. Ігоря Сікорського, Вид-во “Політехніка”, 2021. – 213 с.

18. Демидов, З. Г. Основні види кібератак на WEB-сайти / З.Г. Демідов // Актуальні питання протидії кіберзлочинності та торгівлі людьми : зб. матеріалів Всеукр. наук.-практ. конф. (м. Харків, 15 листоп. 2017 р.) / МВС України, Харк. нац. ун-т внутр. справ; Координатор проектів ОБСЄ в Україні. – Харків : ХНУВС, 2017. –131-134 с.

19. Технології забезпечення безпеки мережевої інфраструктури: підручник / В. Л. Бурячок та ін. К.: КУБГ, 2019. – 218 с.

20. Безменов М. І. Основи програмування у середовищі Delphi : навч. посіб. – Харків : НТУ «ХПІ», 2010. – 608 с.

21. Короткий курс програмування в середовищі Delphi (для студентів заочної форми навчання спеціальності 7.050201 “Менеджмент організацій” спеціалізації “Інформаційні системи в менеджменті”). Укл. Г.А. Мірошніченко. - Харків: ХНАМГ, 2004. - 81 с.

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

22. Бурячок В. Л. Інформаційний та кіберпростори: проблеми безпеки, методи та засоби боротьби: посібник / [В. Л. Бурячок, С. В. Толюпа, В. В. Семко та ін.]. – К. : ДУТ-КНУ, 2016. – 178 с.

23. Сопілко І. М. Інформаційна безпека та кібербезпека: порівняльноправовий аспект// Юридичний вісник Повітряне і космічне право. – Київ: НАУ, 2021. – № 2(59). – 110-115 с.

24. Галіпчак В. Інформаційна війна як складова гібридної війни в умовах російської агресії. Вісник Прикарпатського університету, 2023. Вип. 15. 26-32 с..

25. Чмир Я. Сучасні проблеми інформаційної безпеки України та перспективні напрями їх вирішення». Наукові праці Міжрегіональної Академії управління персоналом. Політичні науки та публічне управління, 2022. №2(62). – 149-154 с.

26. Циріна М Сучасні технології захисту й опрацювання конфіденційної докуметної інформації в організаціях і установах різних форм власності. Національна академія керівних кадрів культури і мистецтв. Бібліотекознавство. Документознавство. Інформологія, 2021. №4. – 40 с.

27. .Керусов М. В. Політика інформаційної безпеки інтернет провайдерів. Кібербезпека в сучасному світі : матеріали ІІ Всеукр. наук.-практ. конф. м. Одеса, 20.11. 2020 р. Одеса: Видавничий дім «Гельветика», 2020. – 27-30 с.

28. Коваленко О.В. Моделі та методи розроблення програмного забезпечення комп'ютерних систем для підвищення безпеки даних: монографія / О.В. Коваленко // К.: Вид. «КОД» – 2019. – 305 с.

29. Коваленко А.В. Методи якісного аналізу і кількісної оцінки ризиків розробки програмного забезпечення / А.В. Коваленко, А.А. Смирнов // Інформаційні технології в управлінні, освіті, науці і промисловості: монографія / Під редакцією професора В. С. Пономаренко. - Х.: Видавець Рожко С. Г., 2016. - 566 с.

30. Коваленко А.В. Розробка методу управління ризиками розробки програмного забезпечення / А.В. Коваленко, А.А. Смирнов // Інформаційні

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

технології : проблеми та перспективи : монографія / За загальною редакцією В. С. Пономаренка. - Х.: Видавець Рожко С. Г., 2017. - 447 с.

31. Коваленко О.В. Методи якісного аналізу та кількісної оцінки ризиків розроблення програмного забезпечення/ О.В. Коваленко // Системи управління, навігації та зв'язку. – Випуск 3 (49). – Полтава: ПолтНТУ. – 2018. – 116-125 с.

32. Коваленко О.В. Оцінка ефективності технології тестування безпеки / О.В. Коваленко // Вчені записки Таврійського національного університету імені В.І. Вернадського. Серія: Технічні науки. Том 29 (68) № 2, 2018. – 137-141 с.

33. Коваленко О.В. Методи та засоби управління безпекою додатків / О.В. Коваленко // Інформаційно-керуючі системи на залізничному транспорті. №4, 2018. – 41-44 с.

34. Коваленко О.В. Розробка інформаційної технології передтестової компіляції та розподілу доступу / О.В. Коваленко // Системи управління, навігації та зв'язку. – Випуск 4 (50). – Полтава: ПолтНТУ. – 2018. – 115-119 с.

35. Коваленко О.В. Аналіз та дослідження інформаційних технологій розробки програмного забезпечення/ О.В. Коваленко // Вчені записки Таврійського національного університету імені В.І. Вернадського. Серія: Технічні науки. Том 29 (68) № 5, 2018. – 131-137 с.

36. Коваленко А.С. Розробка структури бази даних інтегрованої інформаційної системи / А.С. Коваленко, А.В. Коваленко // Інформаційні технології і захист інформації в інформаційнокомунікаційних системах: монографія / Під редакцією професора В. С. Пономаренко. - Х.: Вид-во ТОВ «Щедра садиба плюс», 2015. - 54-64 с.

37. Коваленко А.С. Завдання розпізнавання ситуацій в ERP системах / А.В. Коваленко, А.А. Смірнов, А.С. Коваленко // Системи обробки інформації. - Х.: ХУПС, 2014. - Вип. 4(120). - 161-164 с..

38. 4. Коваленко А.С. Підсистема технічної діагностики для автоматизації процесів керування в інтегрованих інформаційних системах / А.С. Коваленко ,

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

О.А. Смірнов, О.В. Коваленко // Системи озброєння і військова техніка.– Х.: ХУПС, 2014. – № 1(37). – 126-129 с..

39. 5. Коваленко А.С. Аналіз ефективності використання експертної системи технічної діагностики з традиційною структурою / А.С. Коваленко, А.А. Смирнов, А.В. Коваленко // Системи озброєння і військова техніка. - Х.: ХУПС, 2014. - № 2(38). - 106-108 с.

40. Коваленко А.С. Розробка структури експертної системи технічної діагностики інтегрованої інформаційної системи / А.С. Коваленко, А.А. Смирнов, А.В. Коваленко // Наука і техніка Повітряних Сил Збройних Сил України. - Харків: ХУПС, 2014. - № 2(15). - 154-157 с..

41. Коваленко А.С. Розробка структури експертної системи технічної діагностики інтегрованої інформаційної системи / А.С. Коваленко, А.А. Смирнов, А.В. Коваленко // Наука і техніка Повітряних Сил Збройних Сил України. - Харків: ХУПС, 2014. - № 2(15). - 154-157 с.

42. Коваленко А.С. Структура системи технічної діагностики інтегрованих інформаційних систем / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Збірник наукових праць Кіровоградського національного технічного університету / техніка в сільськогосподарському виробництві, галузеве машинобудування, автоматизація. – Кіровоград: Вид-во КНТУ, 2014. – Вип. 27. – 245-251 с.

43. Коваленко А.С. Дослідження будови інтегрованої інформаційної системи та її елементів / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Системи озброєння і військова техніка. – Х.: ХУПС, 2014. – № 4(40). – 85-88 с.

44. . Коваленко А.С. Розробка структури бази даних для обліку технічного стану елементів інтегрованої інформаційної системи з урахуванням вимог споживачів інформації / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Системи обробки інформації. – Х.: ХУПС, 2015. – Вип. 1(126). – 75-79 с.

45. Жилін А. В., Шаповал О. М., Успенський О. А. Технології захисту інформації в інформаційнотелекомунікаційних системах: навч. посіб. Київ: КПІ ім. Ігоря Сікорського, Вид-во “Політехніка”, 2021. – 213 с.

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		94

46. Шуліка К., Балагура Д., Смірнов А., Непокритов Д., Литвин А. Метод використання сучасних систем захисту кінцевих точок (EDR) для убезпечення від комплексних атак». Сучасний стан наукових досліджень та технологій в промисловості. 2024. №2 (28). 1– 82. – 195 с.

47. . Скіцько О. Ширшов Р. Система управління інформаційної безпеки як інструмент підвищення захищеності та ефективності об'єктів критичної інфраструктури. International Science Journal of Engineering & Agriculture. 2023. Vol. 2. No. 6. – 12–22 р.

48. Малькевич Р., Балацька В. Використання SPLUNK для захисту інформації в організаціях. Інформаційна безпека та інформаційні технології: зб. тез доповідей V Всеукр. наук. – практ. конф. молодих учених, студентів і курсантів. – 66–68 с.

49. Richards M. Fundamentals of Software Architecture: An Engineering Approach / M. Richards, N. Ford. – Sebastopol: O'Reilly Media, 2020. – 432 р.

50. Кожанова А.С. Обґрунтування необхідності створення систем технічної діагностики інтегрованих інформаційних систем / О.А. Смірнов, А.С. Кожанова, О.В. Коваленко // Системи обробки інформації. – Х.: ХУПС, 2013. – Вип. 6(113). – 255-257 с.

					ВКРБ-123.26.0011.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		95