

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

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

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи KVM на основі
технології HDBase-T”

Виконав здобувач вищої освіти
II курсу, групи KI-23M
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Поляруш Б.С.
« ____ » _____ 2024 р.

Керівник проекту
доктор філософії (PhD)
_____ Усік П.С.
« ____ » _____ 2024 р.
Рецензент _____

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

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.
Олексій СМІРНОВ
« 6 » вересня 2024 року

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

Полярушу Богдану Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження та програмна реалізація системи KVM на основі технології HDBase-T*

2. Керівник роботи *Усік Павло Сергійович, доктор філософії (PhD)*

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

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

3. Строк подання студентом роботи до захисту *2.12.2024 р.*

4. Мета та завдання випускної кваліфікаційної роботи: *Метою розробки є дослідження та програмна реалізація системи KVM на основі технології HDBase-T*

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

- | | |
|--|--|
| <i>1. Призначення та область використання.</i> | <i>6. Наукова новизна.</i> |
| <i>2. Перегляд аналогічних існуючих систем.</i> | <i>7. Маркетингове та економічне обґрунтування IT-проєкту.</i> |
| <i>3. Опис і обґрунтування проєктних рішень.</i> | <i>8. Заходи з охорони праці та техніки безпеки.</i> |
| <i>4. Етапи програмування системи.</i> | <i>9. Висновки.</i> |
| <i>5. Впровадження системи в промислову експлуатацію</i> | |

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

- | | |
|--|-----------------|
| <i>Наукова новизна</i> | <i>1 аркуш</i> |
| <i>Структурна схема системи</i> | <i>1 аркуш</i> |
| <i>Функціональна схема системи</i> | <i>1 аркуш</i> |
| <i>Діаграма процесів</i> | <i>1 аркуш</i> |
| <i>Блок-схема алгоритму роботи додатку</i> | <i>2 аркуша</i> |
| <i>Показники економічної ефективності</i> | <i>1 аркуш</i> |

6. Консультанти розділів роботи

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

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

КАЛЕНДАРНИЙ ПЛАН

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

Дата видачі завдання
« 6 » вересня 2024 р.

Підпис керівника

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

Підпис здобувача

(прізвище та ініціали)

АНОТАЦІЯ

Поляруш Б.С. Дослідження та програмна реалізація системи KVM на основі технології HDBase-T. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2024.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи KVM на основі технології HDBase-T.

Метою розробки є дослідження та програмна реалізація системи KVM на основі технології HDBase-T.

Об'єктом дослідження є процес KVM на основі технології HDBase-T.

Предметом дослідження є методи KVM на основі технології HDBase-T.

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

Результат роботи – програмна реалізація системи KVM на основі технології HDBase-T.

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

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

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

Програму розроблено в середовищі Python.

Ключові слова: комп'ютерна інженерія, KVM, HDBase-T

ABSTRACT

Polyarush B.S. Research and software implementation of the KVM system based on HDBase-T technology. 123 Computer engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2024.

In this final qualification work for the second (master's) level of higher education, software designed for the KVM system based on HDBase-T technology was developed.

The purpose of the development is research and software implementation of the KVM system based on HDBase-T technology.

The object of the study is the KVM process based on HDBase-T technology.

The subject of research is KVM methods based on HDBase-T technology.

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

The result of the work is a software implementation of the KVM system based on HDBase-T technology.

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

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

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

The program was developed in the Python environment.

Keywords: computer engineering, KVM, HDBase-T

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	6
1.1 Призначення системи.....	6
1.2 Область застосування.....	6
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	8
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	8
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	17
2.3 Розгорнута постановка завдання	20
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	21
3.1 Опис функціонування системи	21
3.2 Розробка структурної схеми.....	23
3.3 Розробка функціональної схеми	33
3.4 Розробка діаграми процесів.....	36
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	38
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	38
4.2 Захист розробленого програмного забезпечення.....	56
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	58
6 НАУКОВА НОВИЗНА	64

						ВКРМ-123.24.0032.00.00.ПЗ		
Вим	Арк.	№ докум.	Підп.	Дата				
Розроб.	Поляруш Б.С.				Дослідження та програмна реалізація системи KVM на основі технології HDBase-T	Літ.	Аркуш	Аркушів
Перев.	Усік П.С.					М	1	91
Н.контр.	Коваленко А.С.				ЦНТУ КІ-23М			
Затв.	Смірнов О.А.							

7	МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ІТ-ПРОЄКТУ	65
7.1	Визначення цільової аудиторії кінцевого готового продукту	65
7.2	Оцінка привабливості шляхом застосування методів експертних оцінок ...	67
7.3	Вибір методу оцінки вартості ПЗ	69
7.4	Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості.....	69
7.5	Пропозиція алгоритму просування проєкту розробки ПЗ	71
7.6	Оптимізація каналів збуту та шляхів реалізації ПЗ	71
7.7	Визначення ключових факторів успіху конкретного проєкту.....	73
8	ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	74
8.1	Вступ.....	74
8.2	Шкідливі і небезпечні фактори при роботі з комп'ютером.....	75
8.3	Аналіз санітарно-гігієнічних умов праці на робочому місці програміста ..	76
8.4	Розробка заходів з умов поліпшення охорони праці	79
8.5	Розрахункова частина	80
9	ОСНОВНІ ВИСНОВКИ.....	83
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	85

КБПЗ-2024

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

- DS – DirectShow мультимедійне розширення Microsoft;
- TrF1 – трансформ-фільтр;
- RnF1 – рендер-фільтр;
- SourceFilter – фільтр джерела;
- IC (Intelligent Connect) – інтелектуальне з'єднання;
- ООП – об'єктно-орієнтоване програмування;
- DSP – MS Direct Show Components and class to write Multimedia Applications – бібліотека компонентів для розробки мультимедійних програм;
- VWINDOW – вертикальна синхронізація;
- HWINDOW – горизонтальна синхронізація;
- PLL – Цикл Блокування Стадії;
- USB – Universal Serial Bus – універсальна послідовна шина;
- URL – universal resource locator – локатор ресурсів Інтернет;
- HTML – HyperText Markup Language – мова розмітки гіпертекстових документів;
- ОС – операційна система;
- ПЕОМ – персональна електронно-обчислювальна машина;
- КС – комп'ютерна система;
- ПЗ – програмне забезпечення;
- ПК – персональний комп'ютер;

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

Актуальність теми. Комп'ютерне встаткування створює шум, виділяє багато тепла й займає чимало місця. Застосування перемикачів і подовжувачів KVM дозволяє рознести обчислювальні блоки й периферію. Розмістивши комп'ютери в серверній кімнаті, можна створити в офісних і інших приміщеннях більше комфортні умови для роботи співробітників, забезпечити необхідні кліматичні параметри для обчислювального встаткування й підвищити рівень захищеності всього рішення.

Така конфігурація широко застосовується в різних центрах керування, демонстраційних залах, навчальних аудиторіях. При цьому все частіше потрібно забезпечити передачу відео у форматі HD без втрати якості.

Потреба в подовженні з'єднань між джерелами й одержувачами сигналу і їхньої комутації виникає при виконанні багатьох завдань. Використання подовжувачів і перемикачів дозволяє ефективно здійснювати моніторинг і керування робітничими середовищами й процесами автоматизації.

У застосовувані для цих цілей сучасних рішеннях знайшли відбиття наступні тенденції й запити:

- перехід від аналогової до цифрової комутації;
- потреба в розподілі цифрового відео;
- гнучкість контролю за периферією й засобами відображення;
- спрощення й здешевлення схем підключення при розширенні функціональності.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи KVM на основі технології HDBase-T.

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

- Огляд існуючих систем KVM на основі технології HDBase-T.

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

- Дослідження системи KVM на основі технології HDBase-T.
 - Програмна реалізація системи KVM на основі технології HDBase-T.
- Об'єктом дослідження є процес KVM на основі технології HDBase-T.*

Предметом дослідження є методи KVM на основі технології HDBase-T.

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

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

- Удосконалено метод KVM на основі технології HDBase-T.
- Розроблено вітчизняний продукт KVM на основі технології HDBase-T,

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

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі KVM на основі технології HDBase-T.

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

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

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи KVM на основі технології HDBase-T, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Поряд із цифровою комутацією, мабуть, найбільш характерною тенденцією є поява й поширення практично необмежене масштабованої матричної комутації. Ці нові цифрові рішення реалізуються на базі як нестандартних спеціалізованих технологій (з використанням витопарної проводки стандартних категорій і волоконних оптичних кабелів), так і стандартних IP-протоколів і комутаторів Ethernet. Вони забезпечують доставку відео високого дозволу в реальному часі без втрат, затримок і рассинхронізації, причому крім розподілу відео дозволяють рознести клавіатуру й мишу, цифрове й аналогове аудіо, послідовні й USB-пристрою.

1.2 Область застосування

Найбільші труднощі представляє подовження відео. На відміну від аналогового сигналу, якість якого зі збільшенням дальності передачі знижується поступово, цифрове відтворення різко переривається після подолання певної відстані. Як і використовуваний у побутових системах HDMI, стандартний цифровий інтерфейс для ПК – DVI – базується на сигналізації TDMS. Застосування нестандартних технологій дозволяє збільшити дальність передачі сигналів HDMI або DVI. Для подовження може використовуватися рідний кабель із повторювачем або витопарний або оптичний кабель із подовжувачем (пари передавач-приймач).

Для передачі незжатого мультимедійного контенту HD-якості по типовій проводці локальної мережі була стандартизована технологія HDBase-T. За рахунок використання складних схем кодування й синхронізації вона дозволяє

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

передавати відео DVI і HDMI, високоякісне аудіо, трафік 100 Base-T, живляча напруга й керуючі сигнали від джерела до приймача по кабелі Категорії 5e й вище на відстань 100 м.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи KVM на основі технології HDBase-T, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

КБПЗ_2024

					VKPM-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

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

Відео – це звичайний ряд записаних один за одним зображень. Як відомо, відео почали записувати за довго до появи першого комп'ютера, тобто в аналоговому виді. Це накладало великі обмеження по його наступній обробці й редагуванні. Аж до початку 1990-х років переважав саме цей тип запису відео. Пізніше були розроблені цифрові технології запису й зберігання відео. Це дало великий поштовх у розвитку не тільки кінематографа, але й домашнього відео. Тобто запис відеороликів і домашніх фільмів стала дуже популярним заняттям. Все це, на тлі розвитку комп'ютерної техніки, змушувало розроблювачів удосконалювати технології запису, зберігання й обробки відеоінформації. От саме для зберігання такої інформації й були розроблені різні відеоформати (на даний момент їхня величезна безліч і орієнтуватися в них дуже складно), які і є основною темою даної статті.

Відео формат визначає структуру відео файлу, тобто те як зберігається файл на носії інформації (CD, DVD або жорсткому диску).

Через великий кількість форматів відео, багато хто стали плутати зовсім різні поняття – «стандарт відео», «відео кодек» і «медіаконтейнер». Давайте розберемося, що ж це за поняття.

Кодек (CODEC = COder+DECoder / кодек = кодер + декодер /) – програмні або апаратні засоби, що перетворюють відеоінформацію в потік ущільнених даних і навпаки.

Медіаконтейнер – формат файлу, що зберігає в собі інформацію про свою внутрішню структуру. Медіаконтейнер фактично є метаформатом, тому що він

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

зберігає дані й інформацію про те, як дані будуть зберігатися безпосередньо усередині файлу.

Можна провести якусь аналогію між медіаконтейнером і zip-архівом. Тобто медіаконтейнери містять відео й аудіо файли усередині себе, як і zip-архів, що містить JPEG-картинку, що буде стисла за допомогою певного алгоритму. За стиск інформації в архівах звичайно відповідає спеціальний модуль, а в медіаконтейнерах роль цього модуля виконує кодек.

Будь-який відео файл має як мінімум дві характеристики визначальну роботу з ним. Це тип медіаконтейнера й кодек (кодеки), якими закодоване його вміст.

Тип медіаконтейнера визначає формат запису різних даних (відео дані, аудіо дані, субтитри, службова інформація й т.д.) у файл. Знаючи тип контейнера програма (наприклад, відео плеєр) може коректно витягати з нього дані, синхронізувати аудіо й відео дані й т.д.

Знання кодека дозволяє програмі із закодованого потоку даних одержати аудіо й відео інформацію втримується в медіаконтейнері.

Популярні кодеки – DivX, Xvid, H.264, Theora, Indeo, x264 і інші.

Популярні медіаконтейнери – AVI, QuickTime, MPEG-4, 3GP і інші.

Нижче розглядаються відео формати по трьох категоріях: аналогове, цифрове й стисле відео. Аналогові формати вже застаріли, тому їхній розгляд було поверхневим. Виключити їх з огляду я не міг, адже потрібно ж мати подання загальної картини. Можете «проскочити» їхній розгляд і перейти відразу до форматів цифрового стислого відео.

Аналогові відео формати

VHS (Video Home System – Домашня Відео Система) – можливості цього формату були обмежені дозволом 240 телевізійних ліній або 320x240 точок на екрані ПК.

Для зйомки використовувалася VHS-касета із шириною стрічки – 12,65 мм і швидкістю – 23,39 мм/с, що мала час запису 240 хвилин при стандартній

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

швидкості стрічки SP (Standard Play) і найнижчу вартість зйомки. Режим запису-відтворення також міг здійснюватися на швидкості LP (Long Play) – 11,695 мм/с.

Головний недолік формату VHS: невисока чіткість зображення (240 ліній по горизонталі) і різка втрата якості при кожному перезаписі. Трохи пізніше, цей відео формат кілька разів валив модернізації. Зокрема, були розроблені формати VHS-Compact (VHS-C) і Super VHS (S-VHS). На даний момент вважається застарілої, в 2008 році припинилося виробництво відеокасет VHS.

Ni8 – «Ni» є скороченням від англійського «High» і має на увазі висока якість зображення, а друга цифра «8» говорить про ширину стрічки в касеті, тобто ширина плівки – 8 мм, швидкість – 28,695 мм/с. Відеодозвіл становить 400 телевізійних ліній (ТВЛ). Аналоговий звуковий потік міг записуватися на касету у двох режимах – моно й стерео. Максимальний час запису становить 180 хвилин у режимі SP (Standard Play) і 360 хвилин у режимі LP (Long Play). Камери забезпечувалися S-Video розніманням для якісного виводу зображення.

Ni8 XR (Ni 8 extended Resolution) – це той же Ni8, але зі збільшеним дозволом, тобто вдосконалена модифікація формату Ni8. Дозвіл зображення досягав 440 ліній по горизонталі при меншому рівні перешкод кольоровості і яскравості за рахунок розширення смуги запису яркісного сигналу в область звукового сигналу.

Обоє вище перерахованих формату (як і VHS) давно застаріли й були замінені цифровими форматами відео.

Цифрові відео формати

DV (Digital video) – в 1993 році компанії Sony, Matsushita(Panasonic), JVS, Hitachi, Mitsubishi, Toshiba, Sanyo, Sharp, Philips і інші створили консорціум Digital Video Cassette (Цифрова відеокасета) або DVC. Пізніше аббревіатура назви скоротилася до DV. Спільними зусиллями ці фірми розробили побутовий цифровий стандарт відеозапису на стрічку шириною 6,35 мм (чверть дюйма). Цей формат забезпечує дозвіл 500 рядків. Оцифровка при записі здійснюється з дозволом 720x576. Відеозображення й звук у форматі DV пишуться роздільно. Це

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

дає можливість додавати звуковий супровід після завершення запису або редагування відео, а також перезаписувати звук. Формат DV забезпечує найвищу якість фото- і відеозйомок і дозволяє зберігати всі дані в цифровому виді на касеті, карті пам'яті або на жорсткому диску комп'ютера.

Mini DV (Mini Digital Video) – цифровий напівпрофесійний формат, створений за рахунок спрощення й здешевлення професійного формату DV. Використовує спеціальні касети Mini DV (ширина стрічки – 6,35 мм, швидкість – 18,831 мм/с), які відтворюються або з камери, або на спеціальному цифровому відеомагнітофоні.

Тривалість запису на одну касету – 60 хвилин SP або 90 хвилин LP. Mini DV має дозвіл зображення до 540 ліній по горизонталі (720x576 точок), підтримується стереозвук якості CD (PCM stereo – 48 кГц/ 16 біт/2 канали або 32 кГц /12 біт/ 4 канали).

Формати стислого цифрового відео

AVI (Audio Video Interleaved) – технологія фірми Microsoft, це найпоширеніший і найменш стислий з відеоформатів. Файли, створені з використанням цього методу, мають розширення .avi. Відео- і аудіодані записуються в один файл на диску в такий спосіб: всі інформаційні потоки розбиваються на безліч рівних частин (chunks) і потім записуються в один файл один за одним по черзі. Спочатку записується заголовок, а потім 1-я частина відео й 1-я частина звуку; потім 2-я частина відео й 2-я частина звуку й т.д. Інакше кажучи, використовується технологія чергування відеокадрів і звуку, який, властиво, і визначається аббревіатура AVI (Audio Video Interleaved). У середньому одна секунда AVI-Зображення займає приблизно 2 Мбайт на жорсткому диску.

Характеристики:

- Тип – медіаконтейнер.
- Сумісні кодеки – DivX, Xvid, Indeo і інші (для відео) і MP3, WMA і інші (для аудіо).

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

– Розширення файлів – .avi.

QuickTime – під «QuickTime» одночасно розуміють і мультимедійний плеєр виробництва компанії Apple, і технологію стислого відео, що надає можливість захоплювати, стискати й програвати широкий спектр цифрового відео- і аудіоконтента. QuickTime включає три основних елементи – QuickTime відеоформат (Movie file format), шар абстрактного носія (Media Abstraction Layer) і набір убудованих медіаслужб. Шар абстрактного носія визначає набір сервісних функцій для створення, редагування й відтворення цифрового матеріалу.

Серед них:

- синхронізація за часом;
- компресія й декомпресія аудіо- і відеоданих;
- перетворення форматів, масштабування, змішування й транскодування;
- аудіо- і відеоефекти й переходи;
- синхронізація читання й запису;
- захват даних;
- імпорт і експорт даних.

Програма й кодеки QuickTime служать для відтворення відео як власного формату (QT і MOV), так і інших найпоширеніших медіаформатів цифрового відео, звуку, тексту, анімації, а також потокового відео з Інтернету. Підтримуються, у тому числі, MP3, AVI, MPEG (включаючи MPEG-4), AAC Audio, Flash.

Характеристики:

- Тип – медіаконтейнер.
- Сумісні кодеки – Apple Video, H.264 і безліч інших (для відео) і Apple Lossless, AAC, MP3, WAV і інші (для аудіо).
- Розширення файлів – .mov, .qt.

Сімейство форматів стиску відео MPEG

Слово «MPEG» є скороченням від Moving Picture Expert Group – назви експертної групи ISO (Міжнародної організації по стандартизації), що діє в

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

напрямку розробки стандартів кодування й стиски відео- і аудіоданих. Часто аббревіатуру MPEG використовують для посилання на стандарти, розроблені цією групою. Технологія MPEG використовує потоковий стиск відео, при якому обробляється не кожний кадр окремо (як це відбувається при стиску відео за допомогою алгоритмів Motion-JPEG), а аналізується динаміка змін відеофрагментів і усуваються надлишкові дані.

MPEG-1 – формат (телестандарт) для зберігання й відтворення відео- і аудіоданих на мультимедіаносіях даних. Якість порівнянна з відеозаписом VHS(дозвіл 352x228 (стандарт PAL) або 320x240 (стандарт NTSC) при частоті 25 або 30 кадрів у секунду відповідно). Формат використовується для запису Video CD. Для кодування відео у формат MPEG-1 використовують спеціальні програми (кодеки).

MPEG-2 і MPEG-3 – стандарт MPEG-2 розроблений як доповнення до стандарту MPEG-1 і підтримує передачу високоякісного відео по високошвидкісних цифрових каналах. Інтенсивність потоку медіаданих до 50 Мбайт/с. У результаті для фільмів, створених у стандартах PAL і SECAM, підтримується дозвіл 720x576 при 25 кадрах у секунду при якості, що практично не уступає віщальному. Формат підходить для всіх стандартів телебачення, широко використовується при записі DVD-дисків. MPEG-1 і MPEG-2 визнані міжнародними стандартами для стиску відео. VMPEG-2, у порівнянні з MPEG-1, додана підтримка багатоканального звуку (Dolby Digital 5.1, DTS і т.п.).

Всупереч очікуванням і прогнозам розробка стандарту MPEG-3 з'явилася всього лише поліпшеною версією MPEG-2. Формат MPEG-3 (не плутайте з форматом стиску аудіоданих MP3) сьогодні став частиною стандарту MPEG-2 і окремо тепер не згадується.

MPEG-4 – формат MPEG-4 задає принципи роботи із цифровим поданням медіаданих для трьох областей: інтерактивного мультимедіа (включаючи продукти, розповсюджені на оптичних дисках і через Інтернет), графічних додатків і цифрового телебачення (DTV). Він має справу не просто з потоками й

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

масивами медіаданих, а з медіаоб'єктами, які є ключовим поняттям цього стандарту.

При транспортуванні відеокартинка розділяється на складені елементи – медіаоб'єкти, описується структура цих об'єктів і їхнього взаємозв'язку, щоб можна було зібрати їх у єдину відеозвукову сцену. Результуюча сцена складається з медіаоб'єктів, об'єднаних в ієрархічну структуру. На відміну від попередніх MPEG-стандартів, які ділили кадр на квадратні блоки поза залежністю від умісту, кодер по стандарті MPEG-4 оперує цілими об'єктами довільної форми. Ключові кадри не розставляються із заданою регулярністю, а виділяються кодером тільки в ті моменти, у які відбувається зміна сюжету. Такі розгалужені алгоритми пошуку й обробки об'єктів складної форми, поглибленого аналізу послідовностей кадрів вимагають істотно більших обчислювальних ресурсів для якісного відновлення (декомпресії) зображення цього формату, ніж у випадку MPEG-1 або 2.

Ефективність компресії відео в MPEG-4Кзволяє розміщати повнометражний фільм тривалістю півтора-друга година з гарною якістю всього на одним стандартом компакт-диску в 700 Мбайт. Однак фільми в MPEG-4 все-таки не дотягають до якості DVD-Відео в стандарті MPEG-2.

MPEG 7 і MPEG 21 – на відміну від попередніх форматів стиску сімейства MPEG, MPEG 7 описує інформацію, представлену в будь-якій формі й не залежить від середовища передачі даних.

Формат стиску MPEG 7 використовує багаторівневу структуру опису аудіо й відео інформації. На вищому рівні прописуються властивості файлу, такі як назва, ім'я творця, дата створення й т.д. На наступному рівні опису формат стиску MPEG 7 указує особливості стисливої аудіо або відео інформації – колір, текстура, тон або швидкість. Однією з відмінних рис MPEG 7 є його здатність до визначення типу стисливої інформації. Якщо це аудіо або відео файл, то він спочатку стискується за допомогою алгоритмів MPEG 1, MPEG 2, MPEG 4, а потім описується за допомогою MPEG 7. Така гнучкість у виборі методів стиску

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

значно знижує обсяг інформації й прискорює процес стиску. Основна перевага формату стиску MPEG 7 над його попередниками складається в застосуванні унікальних дескрипторів і схем опису, які, крім усього іншого, уможливають автоматичне виділення інформації як по загальним, так і по семантичних ознаках, пов'язаним зі сприйняттям інформації людиною. Процедура занесення в каталог і пошуку даних перебувають поза сферою розгляду цього формату стиску.

Розробка формату стиску MPEG 21 – це довгостроковий проект, що називається «Система мультимедійних засобів» (Multimedia Framework). Над розробкою цього формату стиску експерти почали працювати в червні 2000 р. На перших етапах планувалося провести розширення, уніфікацію й об'єднання форматів MPEG 4 і MPEG 7 у єдину узагальнюючу структуру. Малося на увазі, що вона буде забезпечувати глибоку підтримку керування правами й платіжними системами, а також якістю надаваних послуг.

Характеристики

- Тип – сімейство стандартів відео (включає медіаконтейнери й кодеки).
- Сумісні кодеки – H.264 і інші (для відео) і AAC, MP3, WAV і інші (для аудіо).
- Розширення файлів – .mpeg, .mpg, .mpe, .m1v; .dat і .m2v (формати відео VideoCD (VSD) і Super VideoCD(SVCD) дисків – стандарт стиску MPEG-1/2); .vob(формат відео DVD диска, містить відео в MPEG2).

Matroska – проект, націлений на створення відкритого, гнучкого, кроссплатформеного формату мультимедійного контейнера й набору інструментів і бібліотек для роботи з даними в цьому форматі. Це відкритий формат, що розроблялася як альтернатива іншим контейнерам AVI, QuickTime і MPEG. Відмінною рисою цього формату є можливість використання в одному файлі декількох аудіопотоків.

Характеристики:

- Тип – медіаконтейнер.

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

– Сумісні кодеки – H.264 і інші (для відео) і AAC, Vorbis, DTS, MP3 і інші (для аудіо).

– Розширення файлів – .mkv (відео із субтитрами й звуком), .mka (аудіофайли) і .mks(субтитри).

Flash Video – формат відео, використовуваний для передачі даних через Інтернет. Використовується в Youtube, Google Video, RuTube і ін. Популярність цього формату багато в чому пов'язана з тим, що він підтримується Adobe Flash Player 'ом, що поширюється і як кросплатформений самостійний додаток і як браузерний плагін.

Характеристики:

– Тип – медіаконтейнер.

– Сумісні кодеки – H.263 (FLV1), VP6 (FL4), H.264 (FLV5) (для відео) і AAC, MP3(для аудіо).

– Розширення файлів – .flv (як правило).

Як виявилось «розібратися в цьому питанні» (у різноманітті відео форматів) дуже й дуже складно. По-перше, існують десятки різних стандартів відео, які часом сильно відрізняються друг від друга. По-друге, через «війни форматів» з'являються всі нові й нові формати схожі один на одного, але все-таки, що мають свої особливості. По-третє, не все з форматів одержали широке поширення через патенти на їхнє використання й т.п. Все це приводить до тому що зараз досить складно орієнтуватися в цьому різноманітті технологій. Хоча деякі рухи в кращу сторону все-таки є (зокрема з розробкою стандартів MPEG) і, будемо сподіватися, що вони збережуться. Помічу, що я не розглянув у цьому огляді трохи менш популярних (на мій погляд) форматів – медіаконтейнер 3GP (використовується в мобільних телефонах), ASF (Advanced Systems Format) – формат потокового відео, розроблений Microsoft, RealMedia – формат потокового відео фірми «RealNetworks Products and Services».

Також хочеться сказати пари слів про перегляд відео. Часто виникають проблеми при відтворенні того або іншого ролика, якщо в системі немає

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

відповідного кодека для декодування відеопотоку. Це може виникнути через те, що використовується плеєр, що поставляється з кодеками тільки однієї фірми, що не може використовувати чужі кодеки через патентні заборони. Цього можна уникнути встановлюючи так звані кодек-паки. Але я волію універсальні плеєри, у які включені майже всі відомі відеокодеки. Прикладом такого плеєра є The KMPlayer.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Python – високорівнева мова програмування, яку називають другою за популярністю в світі. Її використовують для розробки вебзастосунків, програмного забезпечення, машинного навчання. Python застосовують для вирішення робочих завдань у компаніях Google, Instagram, Facebook, IBM, NASA, Dropbox, Netflix та інших. Розробники цінують цю мову програмування за простоту у вивченні, ефективність та мультиплатформність.

Python – скриптова мова програмування з досить простим синтаксисом. Для розуміння достатньо порівняти принципи написання найпростішої програми, яка виводить на екран текстове повідомлення. Саме тому мова програмування Python більш доступна для новачків, а професіонали встигли адаптувати її для вирішення великої кількості завдань. Це мультиплатформне рішення, тому знання Python дає можливість працювати у різних сферах: від розробки мобільних застосунків до ігрової індустрії та штучного інтелекту.

У мови програмування динамічна типізація: є можливість передавати до функцій будь-який тип даних без попереднього вказання. Інтерпретованість дозволяє знаходити помилки у коді ще до повної збірки у робочий застосунок. При цьому Python дуже чітко дає зрозуміти, де та через що виникла помилка.

Це мова об'єктноорієнтованого програмування (ООП). Програмне забезпечення на Python оформлене у вигляді моделей, які можуть бути зібраними

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

у пакети. Тип та структуру кожного об'єкта можна запитати під час виконання програми. Для кожного з об'єктів можна отримати всю інформацію щодо його внутрішньої структури. Окрім того:

- у мови логічний синтаксис, завдяки чому вихідний код легко читати та розуміти;
- гнучкість та масштабованість Python дозволяє адаптувати високорівневу логіку та розширяти складні застосунки, як тільки виникне така необхідність;
- розробка на Python у більшості випадків проходить швидше, ніж на інших мовах програмування;
- Python – інтерпретована мова програмування. Це значить, що код можна написати у будь-якому текстовому файлі на будь-якій платформі, і потім успішно запусити;
- у Python – колосальна спільнота однодумців. Тож будь-які складнощі конкретних розробників вирішуються колективно.

Проте є декілька особливостей, які можна віднести до недоліків. Це повільність (ця мова програмування хоч і універсальна, проте повільніша за інші), велика кількість ресурсів, необхідних для роботи та «прив'язаність» до системних бібліотек.

Мова програмування Python використовується у наступних сферах:

1. Розробка програмних застосунків будь-якого напрямку.
2. Розробка серверної частини мобільних застосунків (найпопулярніший напрямок).
3. Ігри. Багато сучасних ігор для комп'ютерів (наприклад, World of Tanks) частково чи повністю написані на Python.
4. Вбудовані системи для різних пристроїв. Дуже часто Python використовують для написання внутрішніх платформ управління банкоматами.
5. Скрипти та плагіни до уже реалізованих програм для автоматизації процесів чи створення інших рішень.
6. Тестування (автоматизація цього процесу).

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

7. Машинне навчання. – основна мова для написання алгоритмів і аналітичних застосунків у сфері Machine Learning.

Бібліотеки Python

Різні бібліотеки Python використовують для виконання конкретних завдань. Наприклад, Matplotlib підходить для відображення даних у двовимірній та тривимірній графіці. Pandas підходить для зручної роботи з даними. NumPy дозволяє створювати масиви та керувати ними. Requests використовується для веброзробки. OpenCV-Python відкриває можливості для обробки зображень з метою оптимізації систем «машинного зору».

Найвідоміші фреймворки для мови програмування Python

Фреймворки Python допомагають створити зручне та функціональне середовище для розробки. У них міститься набір інструментів, модулів та бібліотек, корисних для виконання конкретних завдань. Це значно полегшує роботу: наприклад, дає змогу не витрачати час на розписування дій, які повторюються, а використати релевантний інструмент. Тож є можливість позбутися рутинних процесів та сконцентруватися на логіці проєкту.

Серед найпопулярніших фреймворків для Python:

- Django – найстаріший та найвідоміший. Створений для реалізації великих інтерактивних проєктів;
- Pyramid – зручний у налаштуваннях, і дає можливість реалізувати складні нестандартні ідеї;
- Web2py – підходить в першу чергу для вебзастосунків і може використовуватись на будь-яких архітектурах.

Популярні Python IDE

IDE або інтегровані середовища розробки – це програмне забезпечення, яке надає розробникам необхідні інструменти для написання, редагування, тестування та налаштування коду. Для розробки на Python найчастіше використовують IDE PyCharm, IDLE, Spyder та Atom.

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи KVM на основі технології HDBase-T.

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

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

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

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

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

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

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

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

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

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ док.ум.	Підпис	Дата		20

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

HDBase-T в KVM

Технологія передачі високої щільності по кручений парі (High Density Base-T, HDBase-T) знайшла широке застосування в аудіовізуальній галузі завдяки можливості передавати по симетричному кабелі різні типи сигналів: незжаті відео Ultra-HD/3D, 2K/4K і відповідну службову інформацію; аудіо в «важких» форматах; кадри Ethernet зі швидкістю 100 Мбіт/с; керуючі сигнали CEC, RS-232 і інфрачервоні; USB 2.0 і живлення до 100 Вт. При наявності в передавальному й приймаючому пристроях мікросхеми Valens мультимедійні дані в дозволі 4К можна передавати на відстань до 150 м по кабелю крученою парою без використання додаткових пристроїв, що на даний момент є самим передовим способом.

Для передачі відео й інших сигналів KVM більшість постачальників цих пристроїв використовують, як і колись, власні технології, які для багатьох становлять предмет гордості. Так, Raritan підтримує свою технологію обробки відео, що «базується на 30-літньому досвіді компанії». Технологія, розроблена Guntermann & Drunck, дозволяє передавати відео з дозволом до 4К без втрати якості (lossless), а також всі сигнали, включаючи HID, USB 3.0 і RS-232.

Разом з тим всі частіше виробники KVM як базова технологія застосовують у своїх подовжувачах HDBase-T. Наприклад, у новій лінійці відеоматриць Blustream використовується HDBase-T, відзначає Микола Клаптиків, технічний фахівець компанії IMS. У більшості моделей подовжувачів і KVM компанії Rextron використовується HDBase-T. Подовжувачів можуть передавати й трафік Ethernet, і сигнали керування через інфрачервоний порт (крім того, одна з моделей приймача має убудований підсилювач потужності для

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

аудіо). При використанні встаткування з підтримкою HDBase-T замовник не прив'язаний до певного виробника: передавач може бути випущений однією компанією, а приймач – іншою. Головна умова – єдина версія протоколу HDBase-T, що робить дану технологію ще більш гнучкою й зручною в експлуатації. Однак такі подовжувачі, як Pacific X-HDU від Avitech з повною підтримкою HDBase-T, скоріше виключення, чим правило. HDBase-T – дуже гарна й приваблива ідея об'єднати сигнали п'яти різних типів у єдиному інтерфейсі (5 Play). На практиці ж деякі виробники реалізують повну специфікацію 5 Play. Більшість пропонують 3 Play (аудіо, відео, RS-232), а те й 2 Play (аудіо-відео), що, природно, не вирішує завдань KVM.

Стандарт не передбачає строгих вимог до передавачів і приймачів; його специфікації засновані на рекомендаціях. Відповідно, виробники KVM не поспішають відмовлятися від власних пропріетарних технологій на користь HDBaset-T. Крім потенційних проблем із сумісністю, на шляху до широкого застосування HDBase-T для завдань KVM є й інші перешкоди – наприклад, обмеження на дальність, оскільки ця технологія припускає використання кручених пар. KVM всі частіше застосовуються для підключення вилучених комп'ютерів із двома відеовиходами (Dual Head). Як указують в IMS, с допомогою пристроїв на базі HDBT підключити два відеопотоки по одному кабелі не можна, а із застосуванням професійних KVM-подовжувачів – можна. В Extron відзначають, що комутація сигналів HDBase-T – ще більш складне завдання, і кожний виробник змушений вирішувати її по-своєму. Всі ці труднощі ведуть до подорожчання рішення й роблять його неконкурентоспроможним у порівнянні із традиційними KVM.

Таким чином, HDBase-T поки не відповідає повною мірою всім вимогам, які пред'являються до KVM. Можливість передачі USB-сигналів паралельно з відеоінформацією дозволяє використовувати цю технологію в KVM-системах, але це не означає, що потрібно відмовитися від всіх інших, особливо в проектах, де пред'являються особливі вимоги до надійності й відказостійкості. HDBase-T –

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

проміжна технологія на шляху до повноцінної передачі цифрових відеоінтерфейсів по IP.

Для передачі відео на більшу відстань (або по більше вузьких каналах) може використовуватися стиск. У такому випадку сигнал передається по стандартної IP-мережі – як локальної, так і глобальної. Стиск зовсім не припускає низької якості відео – воно може вироблятися без втрати деталей зображення. Разом з тим в інтернет-трансляціях звичайно застосовуються алгоритми, що не уникають втрат, зокрема H.264. Такі алгоритми забезпечують високу якість зображення для звичайного перегляду, але для докладного розгляду важливих деталей, необхідних для фахівців центрів керування або медичних установ, його недостатньо.

Кращі сучасні алгоритми дозволяють передавати по локальній мережі комп'ютерні зображення або повноцінне відео HD-якості в деталях і без яких-небудь візуальних перекручувань. Однак через високі вимоги до пропускну здатності їхнє застосування звичайно не поширюється на глобальні мережі й Інтернет.

3.2 Розробка структурної схеми

Матричні подовжувачі

Матричні перемикачі дозволяють комутирувати різні типи сигналів між безліччю джерел і користувачів. По суті це ті ж подовжувачі сигналів, але в конфігурації «багато хто з багатьма»: джерела й одержувачі сигналу теж підключаються через передавачі й приймачі, при цьому матричний перемикач виступає в ролі свого роду посередника, що дозволяє їм установити з'єднання між собою в довільній комбінації. Поряд з функціональністю KVM вони забезпечують маршрутизацію, комутацію й багатоточковий розподіл цифрових відеосигналів різних форматів високого дозволу.

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

Залежно від числа портів і інших факторів матричні перемикачі мають монолітну або модульну конструкцію. В останньому випадку, як правило, вдається гнучко варіювати спосіб підключення джерел і консолей, вибираючи необхідне середовище передачі – мідь або оптику. Двонаправлені порти перемикача служать як для прийому, так і для передачі сигналу, тобто до них можна підключати як трансівер, так і приймач сигналів. У кращих виробках перемикання між джерелами здійснюється миттєво – протягом одного кадру. Деякі виробники пропонують технології для об'єднання перемикачів, завдяки чому рішення масштабується до декількох тисяч портів.

Для дистанційного підключення периферії до серверів і робочих станцій, коли потрібно одночасно забезпечити передачу відео високої якості, постачальник Interactive Multimedia Solutions (IMS) рекомендує використовувати матричний KVM-перемикач серії DVICenter німецької компанії Guntermann & Drunck. Ці перемикачі забезпечують одночасне керування десятками персональних комп'ютерів. Порти ПК і робочих станцій можуть бути довільно скомбіновані з урахуванням поставлених завдань. Пристрій оснащений резервним блоком живлення, що гарантує його безперебійну роботу у випадку виходу з ладу основного блоку. Рішення забезпечує гнучкі можливості розподілу прав користувачів, для збільшення кількості що підключаються ПК і консольних модулів перемикачі можна каскадувати.

На ринку є моделі з підтримкою сотень портів в одному корпусі. Найбільший на сьогоднішній день перемикач KVM недавно випущений компанією IHSE: у відносно компактному корпусі висотою 25RU розміщається 576 двонаправлених портів. Цей матричний перемикач Draco tera K 480-576 корпоративного класу має архітектуру, що не блокує, і призначений для використання в центрах керування й телевізійних студій. Перемикач має повністю надлишкову системну архітектуру, у тому числі резервуємі джерела живлення, замінні в гарячому режимі модулі й т.д. Міжз'єднання між входом і

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

виходом теж надлишкове: обрив з'єднання між консольним і системним блоками миттєво виявляється, і передача даних здійснюється по резервному каналі.

К 480-576 підтримує відео високого дозволу, у тому числі формати DVI, HDMI, DisplayPort, VGA, HD-SDI і 3 G-SDI (поряд з аналоговими й цифровими аудіоформатами й даними USB 2.0/3.0). При використанні волоконно-оптичних кабелів XV цей перемикач може підтримувати відео UltraHD з дозволом до 4K – 4096x2160. Сьогодні ці потужні пристрої вторгаються в область, де традиційно домінувало таке спеціалізоване встаткування, як відеомікшери, або відеокомутатори (video switcher). Останні розроблялися для перемикання між різними джерелами відеосигналу й забезпечували в числі іншого комбінування або змішання сигналів, саме тому їх часто називають відеомікшерами за аналогією з аудіомікшерами. Разом з тим сучасні моделі відеокомутаторів дозволяють перемикати не тільки відео, але й інтерфейсні HID-пристрої й іншу USB-периферію.

Коли необхідна передача високоякісної комп'ютерної графіки й відео пропонуємо використовувати рішення Crestron Digital Media. Користувачі одержують можливість доступу до будь-якої робочої станції з будь-якого робочого місця. При цьому ще забезпечується вивід інформації на будь-який екран колективного користування. Звичайне подовження USB у системах АВ-комутації обмежується низькошвидкісними пристроями USB 1.1 або підтримкою тільки HID, таких як клавіатура й миша. У рішення DigitalMedia шляхом відновлення прошивання інтегрована технологія SwitchableUSB компанії Icron, завдяки чому стало можливо підключати до нього будь-які USB пристрою, у тому числі стандарту USB 3.0: Web-камери, флеш-накопичувачі й аудіообладнання зі швидкістю передачі даних до 480 Мбіт/с.

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

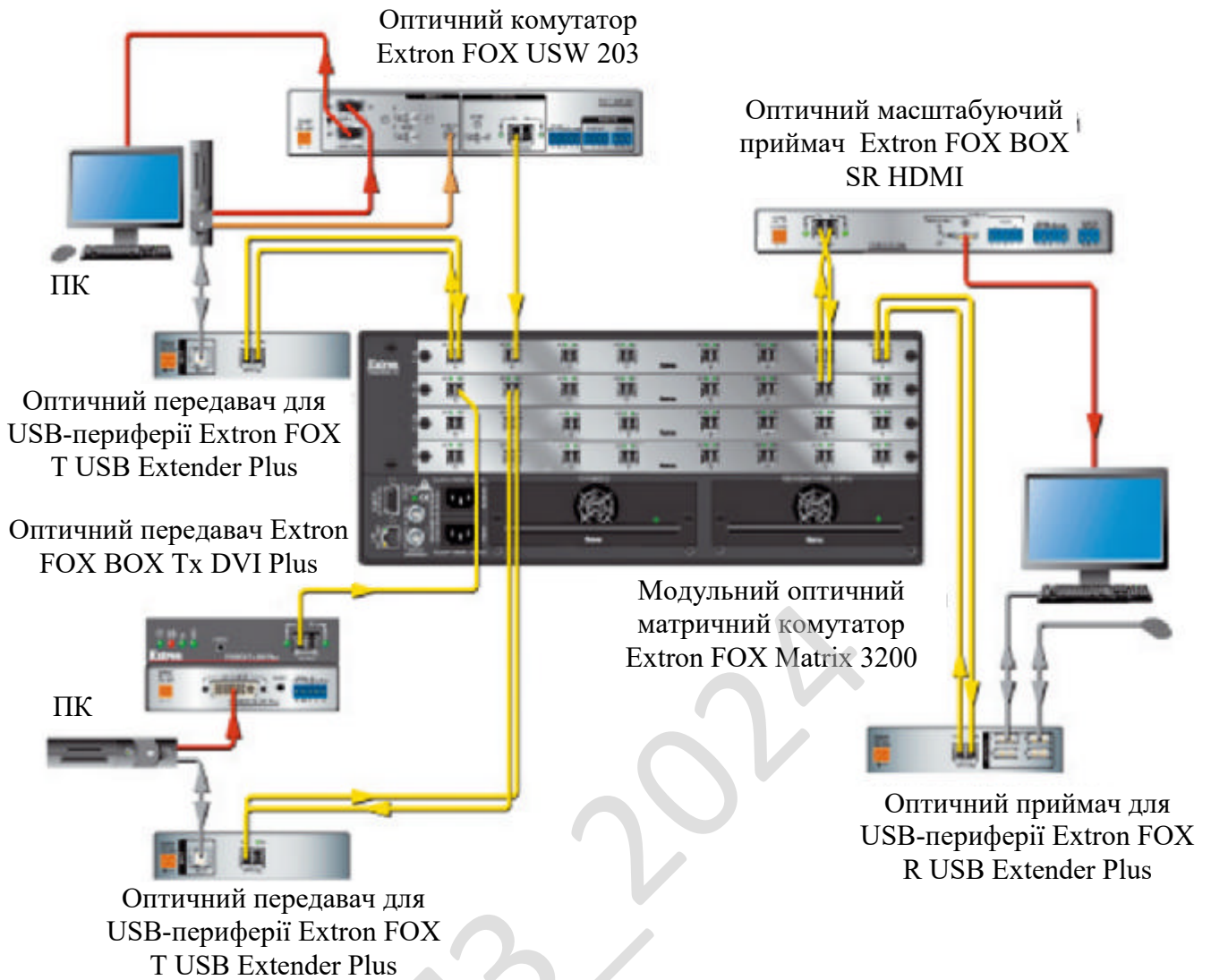


Рисунок 3.1 – Структурна схема системи

Матричний комутатор Crestron DigitalMedia являє собою централізований пристрій, і у випадку виходу його з ладу створюється ризик зупинки всього комплексу. Щоб забезпечити гарантовану працездатність останнього, ми розробили схему з'єднань декількох матричних комутаторів. Завдяки цьому, навіть якщо один із пристроїв DigitalMedia вийде з ладу, користувачі можуть продовжити свою роботу.

Постачальники комутаційного аудіо-відео-устаткування основний акцент роблять на забезпеченні передачі якісного відео, а функціональність KVM розглядається ними як приватна: рішення компанії орієнтовані на побудову

інфраструктури для передачі й обробки сигналів аудіо-відео. Разом з тим практично будь-який подібний проект припускає передачу й комутацію різних аудіо- і відеосигналів від комп'ютерних систем, у тому числі від USB і Ethernet. У лінійці продукції Extron є безліч рішень: від простих АВ-передавачів «точка – точка» по кручений парі або оптичному волокні до складних матричних комутаторів розмірністю тисяча на тисячу входів/виходів. З розрахунком на перспективу була розроблена лінійка матричних комутаторів Extron XTP II Systems із загальною пропускною здатністю 50 Гбіт/с, що дозволяє передавати й комутувати відеосигнали з дозволом до 8К.

Віртуальна матриця

Застосування матричних перемикачів має на увазі створення окремої виділеної інфраструктури. Крім того, сам такий перемикач являє собою спеціалізований пристрій, де, як правило, використовуються нестандартні, пропрієтарні технології. Подібні рішення є у всіх провідних постачальників традиційних KVM. Разом з тим багато хто з них стали розвивати альтернативний підхід, коли для комутації сигналів задіюється наявна локальна мережа й стандартні комутатори Ethernet. Останні для забезпечення ефективної роботи й напрямку відеотрафіку тільки тим одержувачам, хто в ньому зацікавлений, повинні підтримувати протокол IGMP. Щоб у мережі не виникали перевантаження, особливо якщо в ній є присутнім і інший трафік, таке рішення повинне бути ретельно сплановано. Із цієї причини воно має обмежену масштабованість або вимагає створення окремої мережі.

Загальна схема реалізації розподіленої комутуючої матриці загалом однакова у всіх постачальників. До джерел сигналу / системним блокам підключаються передавачі/кодувальники сигналу, а до одержувачів сигналу / периферії – приймачі/декодувальники сигналу. У свою чергу, приймачі й одержувачі приєднуються до комутаторів Ethernet по каналах, пропускна здатність яких повинна бути не менше 1 Гбіт/с. Для збільшення дальності передачі виробники пропонують передавачі й приймачі з оптичними

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

інтерфейсами. Як правило, у мережі встановлюється спеціалізований пристрій або виділений сервер із програмним забезпеченням для керування каналами й правами доступу. Сервер може підключатися в будь-якому місці мережі, при цьому буває досить з'єднання в 100 Мбіт/с.

Крім розподілу високоякісного відео між безліччю одержувачів, таке рішення забезпечує відказостійкість при відповідній конфігурації мережі. Система на основі технології ADDERLink INFINITY дозволяє реалізувати відказостійке рішення на базі стекових комутаторів – при виході з ладу одного з них інші беруть на себе обслуговування трафіку. Це рішення затребуване там, де неприпустимий навіть незначальний простій. Однак є й обмеження, наприклад по кількості одночасно підключених пристроїв. Система передбачає нарощування кількості пристроїв, але при цьому зростає обсяг трафіку будуть позначатися на якості відображення інформації й частоті розгорнення. Тому така система стійко працює при відносно невеликій кількості робочих місць.

Компанія Aten пропонує систему Aten KE System на базі KVM-подовжувачів KE6900 і KE6940. Кожний такий подовжувач складається з пари рознесених пристроїв – приймача й передавача; вони підключаються по локальній мережі Gigabit Ethernet, утворюючи, як затверджується, необмежений віртуальний матричний KVM-перемикач із можливістю формування відеостіни. Крім подовжувачів, до складу рішення входить програмне забезпечення ССКМ. Завдяки убудованому високопродуктивному графічному процесору подовжувачі забезпечують стабільну, без затримок передачу й відтворення зображення з дозволом 1920x1200 пікселів. Технологія Fast Switching дозволяє операторові за вилученою консоллю миттєво перемикатися між джерелами з різним дозволом.

До одного приймача може бути підключено кілька передавачів по різних інтерфейсах, у результаті сформована система здатна контролювати по KVM один сервер, управляти по RS232 іншим, підключитися по USB до третього, слухати й передавати(мікрофон) звук на четвертий. Крім того, устаткування можна поєднувати (агрегувати) у кілька блоків під єдиною консоллю керування з

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

підключенням одночасно декількох моніторів на одному робочому місці оператора, що значно розширює можливості, наприклад операторів ситуаційних центрів. Для захисту переданих по мережі даних подовжувач KE задалегідь кодує їх за допомогою 128-розрядного шифрування AES.

Крім Adder і Aten, схожими рішеннями розташовують і інші виробники KVM. Так, Raritan пропонує систему розподілу аудіо й відео по IP Raritan AV-over-IP Distribution System (RAV-IP). Для конфігурації використовується графічний інтерфейс на базі Web. Перемикання каналів уведення-виводу може здійснюватися за допомогою пульта дистанційного керування, як на звичайних телевізорах. Компанія позиціонує цей продукт як, що розгортається швидко систему, для невеликих інсталяцій, де потрібно забезпечити подовження, розподіл і комутацію високоякісного аудіо й відео. Подібно іншим таким же рішенням, RAV-IP призначена для застосування в самих різних сценаріях, включаючи Digital Signage і відображення мультимедійного контенту в магазинах, готельних лобі, приймалень покоях, кімнатах для нарад, центрах керування й т.д.

Крім потенційних обмежень у масштабуванні, у цих рішень є й інші недоліки, які накладають обмеження на їхнє використання. Як відзначає Микола Клаптиків, технічний фахівець компанії IMS, для створення IP-матриці потрібно додатково здобувати досить дороге мережне встаткування, через що витрати на реалізацію проекту збільшуються й стають порівнянні з вартістю побудови волоконно-оптичної KVM-матриці, якість передачі сигналів у якій на порядок вище. Явним мінусом є й імовірність затримок при поширенні сигналів, що неприпустимо для багатьох організацій: військових підприємств, диспетчерських, студій теле- і радіомовлення в прямому ефірі та ін.

Відео окремо, керування окремо

Гібридна схема з роздільними середовищами для передачі відео й керуючих сигналів клавіатури й миші використовується в рішенні компанії RGB Spectrum: для доставки відео задіється виділена інфраструктура, а для

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

пересилання керуючих сигналів – локальна мережа (по міркуваннях безпеки теж найчастіше виділена).

Такий підхід пов'язаний з тим, що для передачі й комутації якісного відео потрібні виділені ресурси. Інтегрована KVM-система Enterprise MultiPoint Control Room Management System (Enterprise MCMSTM) від RGB Spectrum дозволяє групі користувачів управляти роботою безлічі комп'ютерів і джерел даних, при цьому вона забезпечує вивід зображень від джерел на персональні монітори співробітників і на екран відеостіни в багатовіковому режимі.

Центральним елементом KVM-системи RGB Spectrum є комутаційне встаткування. За доставку високоякісних відеосигналів від джерел до засобів відображення відео в системі Enterprise MCMSTM відповідають професійні матричні комутатори LINX і OPTO. Сигнали можуть передаватися як по кручений парі, так і по оптиці. У першому випадку використовуються пристрої CAT LINX на базі технології HDBase-T, у другому – оптичні передавачі й приймачі серії OPTO. Вивід зображень у багатовіковому режимі на екран відеостіни забезпечується процесорами відеостін серій MediaWall і MediaWall V. Останні підтримують джерела відеосигналу з дозволом 4K і вивід зображень на 4K-дисплеї. Робота всіх пристроїв (матриць, процесорів відеостін і т.д.) і призначення прав операторів контролюються спеціальним сервером.

Сигнали керування в системі Enterprise MCMSTM передаються по окремій замкнутій локальній мережі IP. Для забезпечення безпеки передачі даних вона може бути ізольована від зовнішніх мережних підключень і здійснювати передачу команд керування тільки між пристроями RGB Spectrum, що входять до складу Enterprise MCMSTM, і ПК-джерелами, якщо останнє не заборонено на конкретному об'єкті оснащення. Таким чином, сигнали клавіатур і мишей, підключених до робочих станцій операторів, передаються по IP-мережі Enterprise MCMSTM. Вони будуть передані джерелу, тільки якщо оператор зареєстрований на сервері й має права на керування даним джерелом.

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

Дистанційне з'єднання з комп'ютером може бути встановлено двома способами: за допомогою встановленого на комп'ютері програмного агента (Remote Desktop Agent, RDA) або через зовнішній адаптер (External Desktop Adapter, EDA), запатентований компанією RGB Spectrum. Адаптер EDA – це апаратний пристрій, що працює на фізичному рівні моделі OSI і забезпечує односпрямовану передачу сигналів керування. EDA одержує сигнали від клавіатури й миші через мережу Enterprise MCMSTM і відсилає їхньому комп'ютеру-джерелу через порт PS/2 (допускається використання адаптера PS/ 2-to-USB). Оскільки порт PS/2 односпрямований, команди можуть бути передані на комп'ютер, але одержати доступ до даних комп'ютера або скомпрометувати їх неможливо. Таке рішення особливо ефективно, коли потрібно забезпечити високий рівень безпеки й/або мережної ізоляції ПК-джерела. Завдяки всім цим обережностям система Enterprise MCMSTM може працювати як у захищених, так і незахищених мережах.

За межами локальної мережі

Сучасні системи для комутації сигналів KVM здатні забезпечити передачу по локальній мережі відео високого дозволу до 1920x1200 пікселів (і навіть вище) із частотою кадрів до 60 Гц при кольоровості глибиною до 24 біт. Однак якщо доступ до керованого встаткування треба надати для тих, хто перебуває за межами локальної мережі, якістю відео прийде поступитися. При цьому головна проблема скоріше не технічна (хоча питання затримки коштує ще більш гостро), а економічна: створення виділеної мережі з необхідною пропускну здатністю обійдеться занадто дорого. На даний момент передача високоякісного відео по мережах загального користування нереалізована, тому що потрібне виділення гарантовано високої пропускну здатності.

За межами мережі, виділеної для критично важливих додатків, важко гарантувати передачу високоякісного відео. Якщо для з'єднань «точка – точка» ще їсти шанс забезпечити належну якість, то для багатоточечних з'єднань це проблематично. У тих же ситуаціях, коли безперебійної роботи

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

використовуваних додатків не потрібно, для передачі відео можуть задіятися й Інтернет-З'єднання. Проте навіть у цьому випадку необхідно передбачити широкий канал доступу в Інтернет, оскільки для відео високого дозволу потрібна «гарна пропускна здатність».

Як правило, виробники KVM більше оптимістичні відносно можливостей і перспектив IP KVM, чим постачальники систем для комутації аудіо-відео. IP KVM – гнучка технологія, що може бути зконфігурована для використання в локальних і територіально розподілених мережах, і навіть у глобальні. Деяким компаніям, наприклад віщальним, можуть знадобитися рішення, де стиск і затримки зовсім неприпустимі, так що їм необхідні більше дорогі (і при цьому менш гнучкі) рішення на базі виділеної мережі з оптичною проводкою. Разом з тим IP KVM здатні, на його думку, забезпечити від 90 до 100% вимог до відео з боку більшості компаній, у тому числі до високопродуктивного відеодоступу, реалізованому на основі більше гнучкого, простого в розгортанні й економічному рішенні.

Здатність IP KVM задовольнити необхідні вимоги до передачі високоякісного відео залежить не тільки від правильності схеми використання й підбора встаткування, але й від рівня розвитку мережної інфраструктури, у якій будуть застосовуватися ці схеми, а також від грамотного розподілу її ресурсів. При поширенні цих завдань за межі локальної мережі перераховані умови поширюються й на зовнішню інфраструктуру. На сьогоднішній день IP KVM здатні задовольнити вимоги до передачі високоякісного відео в локальній мережі, а в міру розвитку зовнішньої мережної інфраструктури це стане можливо й за її межами.

Таким чином, застосування сучасних рішень для подовження цифрових відеосигналів і периферії дозволяє реалізувати безліч конфігурацій для самих різних завдань, де потрібно організувати дистанційне підключення пристроїв для керування й відображення, установлених у ситуаційних центрах, телевізійних студіях, переговорних кімнатах, навчальних аудиторіях, медичних установах і т.д.

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

При цьому технології й рішення на базі KVM всі частіше використовуються для виконання традиційних завдань із області АВ, таких як Digital Signage, і навпаки. У результаті споживачі одержують широкий вибір можливих продуктів, що відрізняються за ціною, характеристикам і функціональності.

3.3 Розробка функціональної схеми

На рисунку 3.2 зображена функціональна схема системи. Нижче розглянемо її більш докладно. Для того, щоб зрозуміти, як функціонально працює реалізація KVM системи на основі технології HDBase-T розроблені наступні алгоритми:

– Алгоритм збільшення частоти кадрів відео потоку на основі піксельної компенсації руху та подальше придушення ефекту тремтіння кадру у VideoStream.

– Алгоритмів придушення тремтіння кадру у VideoStream з використанням Image-Based Rendering.

Наведемо спершу опис цих алгоритмів.

Розроблений алгоритм дозволяє збільшувати частоту кадрів VideoStream в довільне ціле число разів. Перетворення частоти проводиться за рахунок збільшення загального числа кадрів VideoStream – додавання нових (інтерпольованих) кадрів. Нові кадри вставляються у вихідну відео послідовність таким чином, що число інтерпольованих кадрів між будь-якими двома сусідніми вихідними кадрами одне і те ж і пропорційно кратності.

Розглянемо як він працює. Одна ітерація алгоритму починається з обчислення карти відео векторів далі проводиться сегментація даних та обчислення інтерпольованих кадрів. З правого боку описано основну формулу алгоритму.

Ефект тремтіння кадру виникає через неякісну лінію зв'язку з мережею Інтернет. Спочатку проходить визначення яскравості пікселя.

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

Далі проходить розрахунок схеми розташування локальних сусідів пікселя та схеми розташування глобальних сусідів пікселя з обчисленням координат поточного пікселя на інтерпольованому кадрі з врахуванням вектора руху у наведеній формулі.

Основне призначення системи це реалізація реалізації KVM системи на основі технології HDBase-T для обробки відео потоку застосовуючи алгоритм піксельної компенсації руху.

Функціональна схема розробленої системи зображена на рисунку 3.2. З рисунку видно, що розроблена система працює на алгоритмі збільшення частоти кадрів відео потоку на основі піксельної компенсації руху та подальше придушення ефекту тремтіння кадру у VideoStream.

Програмний продукт, забезпечує універсальну систему обробки вхідного потоку, забезпечує гарантовано стійкій відео сигнал з мінімальною кількістю шумів та з максимальною різкістю зображення яка ґрунтується на нових алгоритмах обробки відеозображень та частково модернізованих існуючих.

Розглянемо її, спочатку ми отримуємо вхідні дані – цифровий потік відео інформації як він є, тобто неякісний, нечіткий. Він має N кількість кадрів.

До ОС Windows 8 він потопляє через підсистему драйверів (WINDOWS DRIVER MODEL).

Спочатку до драйверу відео пристрою потім до блоку управління WDM пристроями далі до WDM драйверу операційної системи.

Після проходження такої ланки розроблене ПЗ може його обробляти що і проходить.

Розроблене ПЗ приймає відео потік та застосовує наступні алгоритми:

- Проведення просторово-часового придушення шуму у VideoStream.
- Стиск відео потоку на основі алгоритму Video Codec Multi-Scale Retinex (MSR).
- Збільшення частоти кадрів VideoStream на основі піксельної компенсації руху.

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

– Відстеження точкових особливостей у VideoStream, зміна різкості та придушення ефекту тремтіння кадру.

В кінцевому випадку на виході ми отримуємо оброблений відеопотік зі збільшеною кількістю кадрів $N+A$.

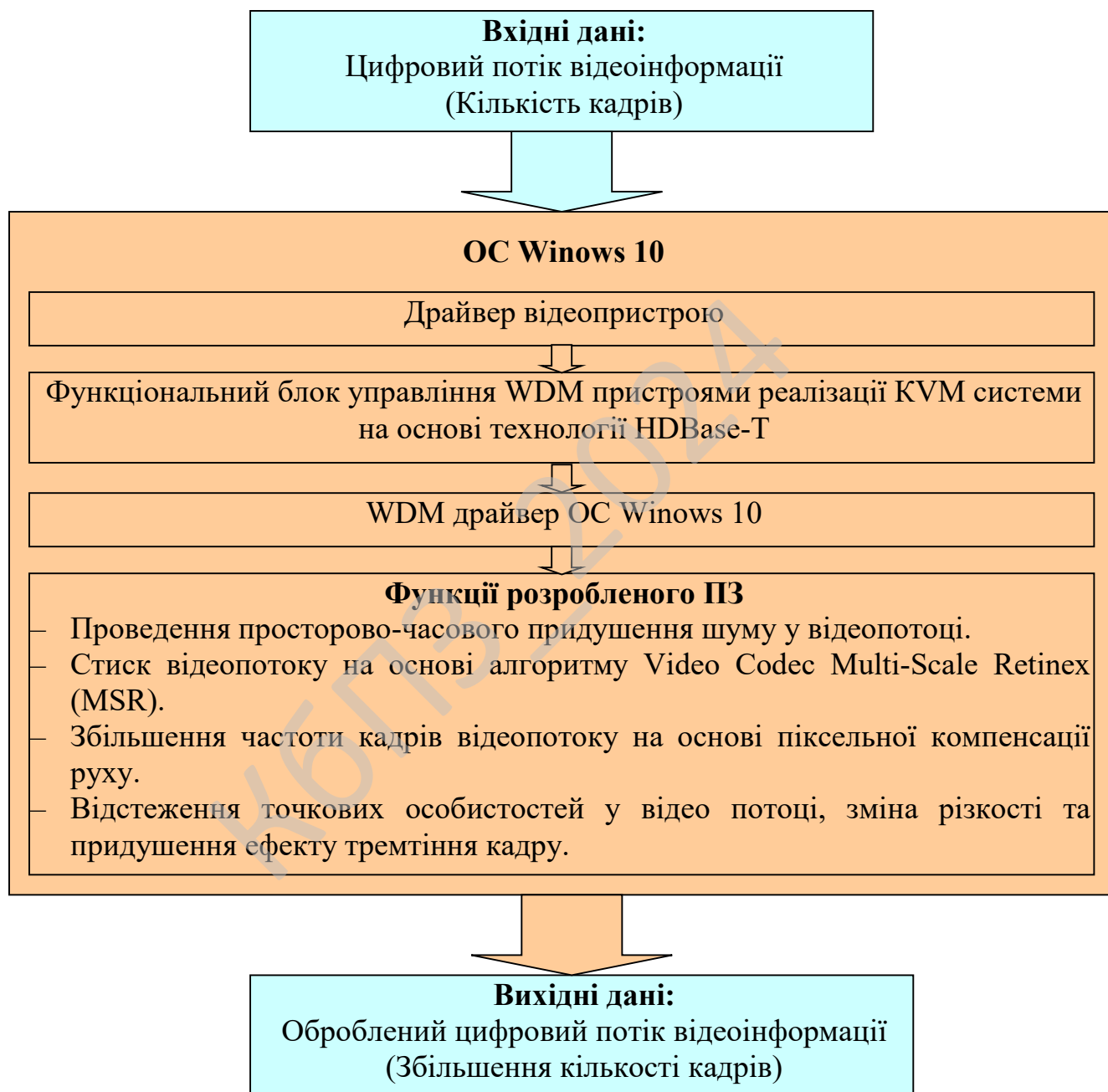


Рисунок 3.2 – Функціональна схема системи

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

3.4 Розробка діаграми процесів

Діаграма взаємодії процесів використовується для візуалізації процесів обробки даних (структурне проектування). Для розробника вважається звичним спочатку креслити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи.

Ця діаграма в подальшому підлягає уточненню шляхом деталізації процесів та потоків даних з метою показати систему що розробляється. Діаграма процесів розробленої системи зображена на рисунку 3.3. При детальному її розгляді можна побачити як саме проходить взаємодія у розробленій системі. Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі.



Рисунок 3.3 – Діаграма взаємодії процесів

Діаграми потоків даних містять чотири типи елементів:

– Процеси які являють собою трансформацію даних в рамках описуваної системи.

– Сховища даних (репозиторії).

– Зовнішні по відношенню до системи сутності.

– Потоки даних між елементами трьох попередніх типів.

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

КБПЗ_2024

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

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

При виборі початкової точки відліку при побудові схем було враховано, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об'єктно-орієнтована що вимагає оптимізації, також те, що при розробці програми слід надати особливу увагу модулю KVM системи на основі технології HDBase-T.

Функціональні блоки на схемі позначають прямокутниками, всередині яких надписують їх найменування відповідно до функцій, що виконуються. Зв'язки між функціональними блоками (внутрішні впливи) позначаються лініями зі стрілками, які вказують напрям впливів.

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

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

З яких видно що робота основної програми складається з початкових етапів ініціалізації ПЗ, перевірки наявності ресурсів системи, блоку початку основного циклу з чеканням запиту від користувача в якому відбувається виклик підпрограми та останньої стадії – перевірки поточного стану та поверненням на початок схеми чи з завершенням роботи розробленого ПЗ.

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

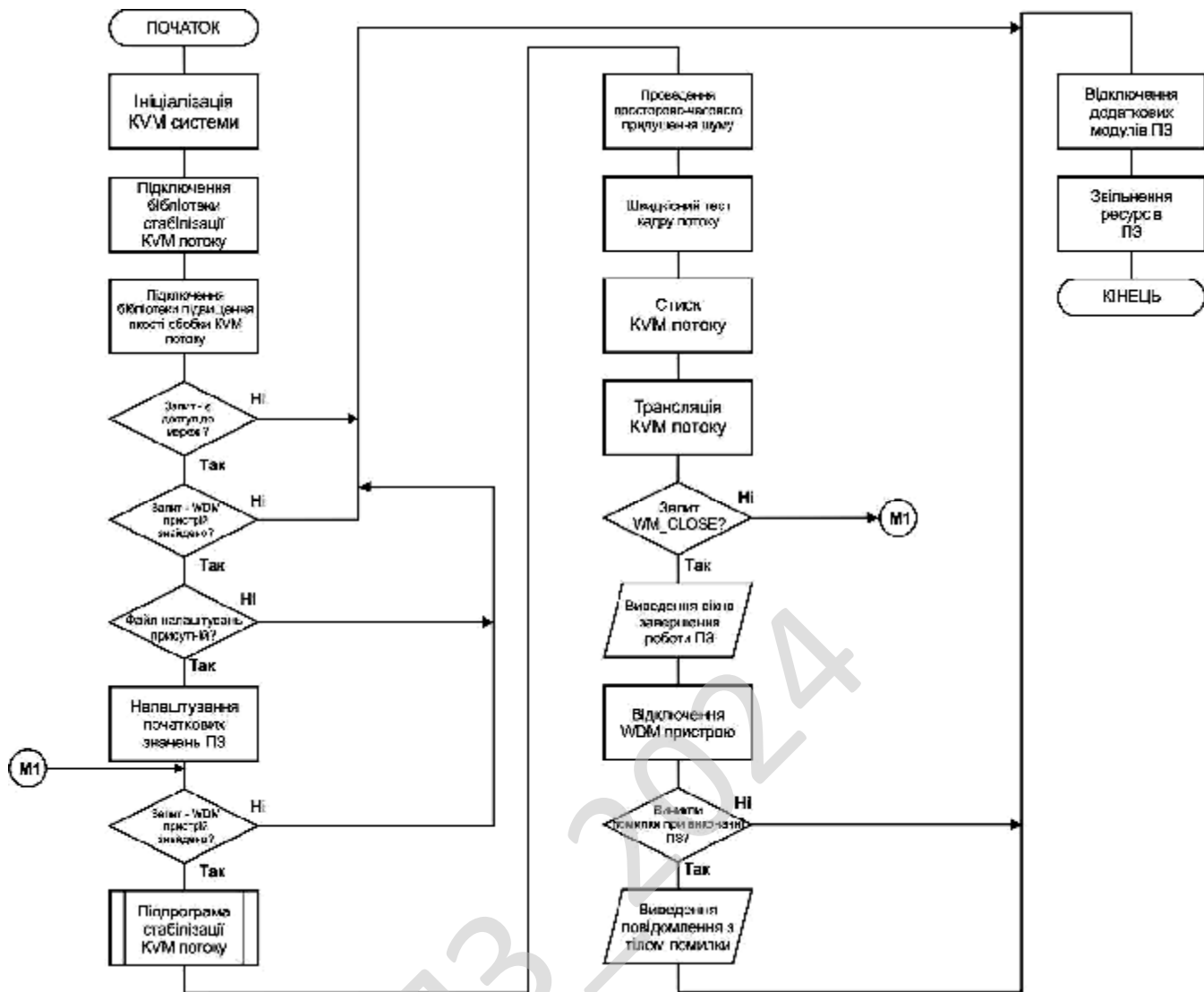


Рисунок 4.1 – Блок-схема основної програми

При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання.

Було використано підходи з використанням UML, це уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування. Є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, називаної UML-моделлю. UML був створений для визначення,

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

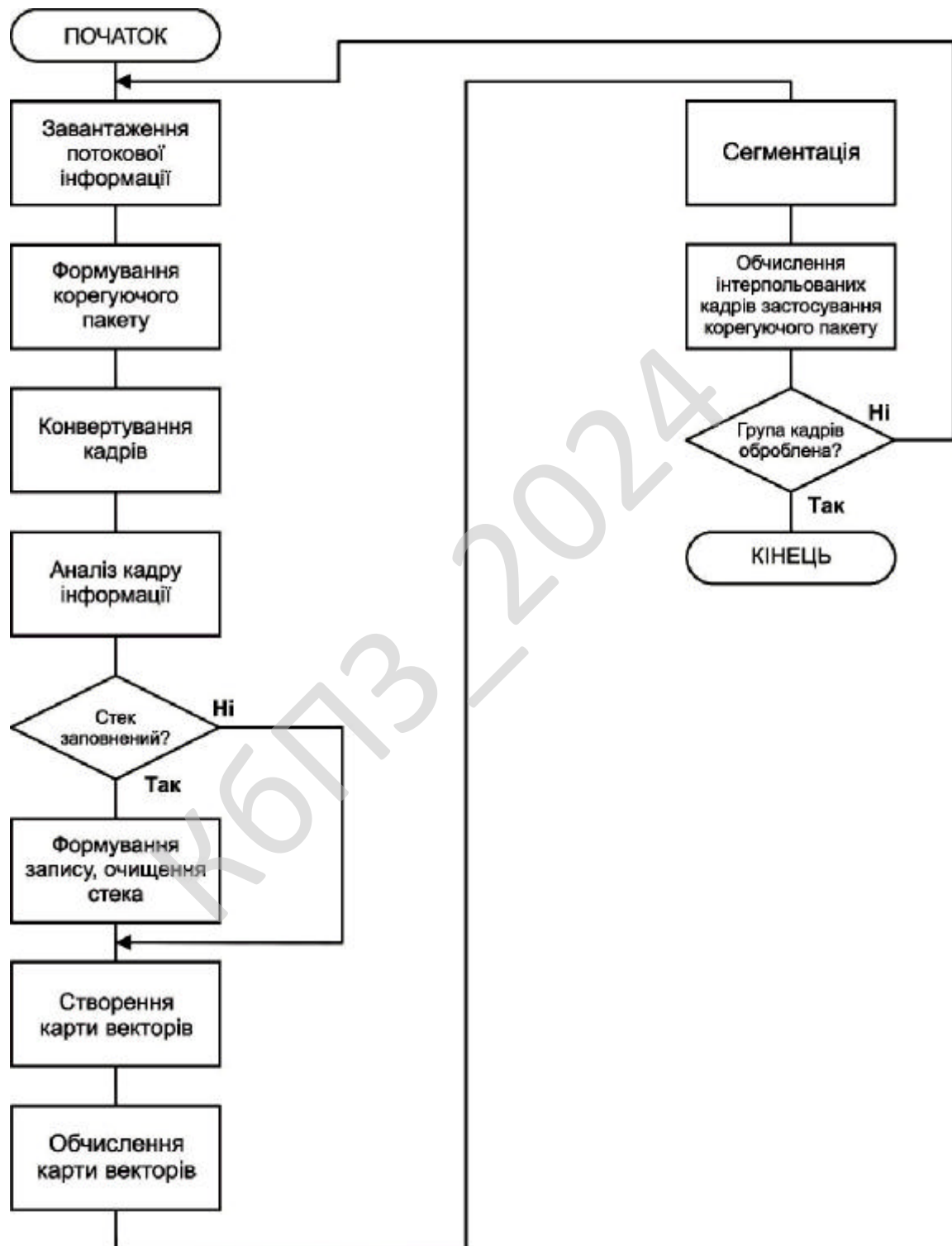


Рисунок 4.2 – Блок-схема роботи підпрограми

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

Архітектура клієнт-сервер є одним із архітектурних шаблонів програмного забезпечення та є домінуючою концепцією у створенні розподілених мережних програм і передбачає взаємодію та обмін даними між ними. Вона передбачає такі основні компоненти:

- набір серверів, які надають інформацію або інші послуги програмам, які звертаються до них;
- набір клієнтів, які використовують сервіси, що надаються серверами;
- мережа, яка забезпечує взаємодію між клієнтами та серверами.

Сервери є незалежними один від одного. Клієнти також функціонують паралельно і незалежно один від одного. Немає жорсткої прив'язки клієнтів до серверів. Більш ніж типовою є ситуація, коли один сервер одночасно обробляє запити від різних клієнтів; з іншого боку, клієнт може звертатися то до одного сервера, то до іншого. Клієнти мають знати про доступні сервери, але можуть не мати жодного уявлення про існування інших клієнтів.

Дуже важливо ясно уявляти, хто або що розглядається як «клієнт». Можна говорити про клієнтський комп'ютер, з якого відбувається звернення до інших комп'ютерів. Можна говорити про клієнтське та серверне програмне забезпечення. Нарешті, можна говорити про людей, які бажають за допомогою відповідного програмного та апаратного забезпечення отримати доступ до тієї чи іншої інформації.

Загальноприйнятим є положення, що клієнти та сервери – це перш за все програмні модулі. Найчастіше вони знаходяться на різних комп'ютерах, але бувають ситуації, коли обидві програми – і клієнтська, і серверна, фізично розміщуються на одній машині; в такій ситуації сервер часто називається локальним.

Модель клієнт-серверної взаємодії визначається перш за все розподілом обов'язків між клієнтом та сервером. Логічно можна відокремити три рівні операцій:

– рівень представлення даних, який по суті являє собою інтерфейс користувача і відповідає за представлення даних користувачеві і введення від нього керуючих команд;

– прикладний рівень, який реалізує основну логіку ПЗ і на якому здійснюється необхідна обробка інформації;

– рівень управління даними, який забезпечує зберігання даних та доступ до них.

Дворівнева клієнт-серверна архітектура передбачає взаємодію двох програмних модулів – клієнтського та серверного. В залежності від того, як між ними розподіляються наведені вище функції, розрізняють:

– модель тонкого клієнта, в рамках якої вся логіка ПЗ та управління даними зосереджена на сервері. Клієнтська програма забезпечує тільки функції рівня представлення;

– модель товстого клієнта, в якій сервер тільки керує даними, а обробка інформації та інтерфейс користувача зосереджені на стороні клієнта. Товстими клієнтами часто також називають пристрої з обмеженою потужністю: кишенькові комп'ютери, мобільні телефони та ін.

Типовим прикладом клієнт-серверної взаємодії є WWW. Існує величезна кількість веб-серверів, на яких розміщується та чи інша інформація. У найпростішому випадку ця інформація являє собою набір веб-сторінок, які можуть зберігатися на сервері у вигляді файлів, розмічених за допомогою мови розмітки HTML. Але ситуація, як правило, є складнішою; значна частина веб-ресурсів на сучасному етапі є динамічними, тобто вони не існують в заздалегідь підготовленому вигляді, а створюються безпосередньо в процесі обробки запиту від користувача.

Для того, щоб людина, яка працює в Інтернеті, могла переглянути ту чи іншу сторінку, на її комп'ютері повинно бути встановлено відповідне програмне забезпечення. Програми для перегляду веб-сторінок називаються браузером.

Але, крім браузерів, до серверів можуть звертатися і інші клієнти, а саме – автономні програми. Вони можуть передбачати взаємодію з людиною, а можуть працювати в цілком автоматичному режимі. Типовим класом таких програм є роботи, призначені для автоматичного перегляду веб-ресурсів. Зокрема, роботи є важливим елементом пошукових систем і використовуються ними для перегляду сторінок і збору інформації про них.

Для запиту до веб-сервера клієнтська програма повинна задати місцезнаходження комп'ютера, на якому розміщується серверна програма, назву потрібного документа і, можливо, інші дані, які специфікують запит. Мережа забезпечує знаходження сервера і передачу йому клієнтського запиту. Серверні програми обробляють цей запит, відповідь пересилається по мережі клієнтові.

Трирівнева клієнт-серверна архітектура, яка почала розвиватися з середини 90-х років, передбачає відділення прикладного рівня від управління даними. Відокремлюється окремий програмний рівень, на якому зосереджується прикладна логіка ПЗ. Програми проміжного рівня можуть функціонувати під управлінням спеціальних серверів ПЗ, але запуск таких програм може здійснюватися і під управлінням звичайного веб-сервера. Нарешті, управління даними здійснюється сервером даних.

Для роботи з системою користувач використовує стандартне програмне забезпечення –звичайний браузер. Це позбавляє його необхідності завантажувати та інсталювати спеціальні програми (хоча інколи така необхідність все-таки виникає).

Але користувачеві слід надати в розпорядженні інтерфейс, який дозволяв би йому взаємодіяти з системою і формувати запити до неї. Форми, що визначають цей інтерфейс, розміщуються на веб-сторінках та завантажуються разом з ними.

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

Веб-оглядач формує запит та пересилає його до сервера, який здійснює обробку. При необхідності сервер викликає серверні програмні модулі, які забезпечують обробку запиту і в разі потреби звертаються до сервера даних. Сервер даних здійснює операції з даними, що зберігаються в системі та складають її інформаційну основу. Зокрема, він може здійснити вибірку з інформаційної бази відповідно до запиту та передати її модулю проміжного рівня для подальшої обробки. Дані, з якими працює сервер даних, найчастіше організовані як реляційна база даних.

Найчастіше веб-сервер і серверні модулі проміжного рівня розміщуються на одному комп'ютері, хоч і являють собою окремі і логічно незалежні програмні модулі.

На сучасному етапі для програмування модулів проміжного рівня використовується мова серверних сценаріїв PHP, а для управління даними – СУБД MySQL. Таким чином, зв'язку PHP-MySQL слід розглядати як стандартний інструмент для створення порівняно простих інтерактивних веб-сайтів та систем електронної комерції; близько 90% комерційних систем сьогодні створюється саме на цій основі. Водночас як засоби управління даними, так і middleware-засоби можуть бути найрізноманітнішими. Так, для створення серверних програм, крім PHP, широко застосовуються Java, Perl, Python, Python.

Взагалі, технології створення розподілених, зокрема веб-програм, стрімко розвиваються. Слід згадати про технології EJB (Enterprise Java Beans), CORBA, а також про .NET – порівняно нову ініціативу компанії Microsoft. Для зберігання даних та їх передачі часто використовується так звана розширювана мова розмітки XML (Extensible Markup Language).

При розробці ПЗ було використано підходи ризик-менеджменту – це система управління ризиками, яка включає в себе стратегію та тактику управління, направлені на досягнення основних цілей. Ефективний ризик-менеджмент включає:

– систему управління;

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

- систему ідентифікації і вимірювання;
- систему супроводження (моніторингу та контролю).

Сучасна наука представляє ризик як вірогідну подію, в результаті настання якої можуть відбутися позитивні, нейтральні або негативні наслідки. Якщо ризик припускає наявність як позитивних, так і негативних результатів, він відноситься до спекулятивних ризиків. Якщо ж наслідки негативні, або відсутні взагалі, такий ризик іменується чистим.

Мета ризик-менеджменту – підвищення конкурентоспроможності господарюючих суб'єктів за допомогою захисту від реалізації чистих ризиків.

Теорія ризик-менеджменту ґрунтується на трьох базових поняттях: корисності, регресії і диверсифікації.

У 1738 швейцарський математик Даніель Бернуллі доповнив теорію вірогідності методом корисності або привабливості того або іншого результату подій. Ідея Бернуллі полягала в тому, що в процесі ухвалення рішення люди приділяють більше уваги розміру наслідків різних результатів, ніж їх вірогідність.

В кінці XIX століття англійський дослідник Ф. Гальтон запропонував вважати регресію або повернення до середнього значення універсальною статистичною закономірністю. Суть регресії трактувалася ним як повернення явищ до норми з часом. Згодом було доведено, що правило регресії діє в найрізноманітніших ситуаціях, починаючи з азартних ігор та розрахунку вірогідності виникнення нещасних випадків, і закінчуючи прогнозуванням коливань економічних циклів.

У 1952 аспірант Університету Чикаго Гарі Марковіц в статті «Диверсифікація вкладень» («Portfolio Selection») математично обґрунтував стратегію диверсифікації інвестиційного портфеля, зокрема, він показав, як шляхом продуманого розподілу вкладень мінімізувати відхилення прибутковості від очікуваного показника. У 1990 Г. Марковіцу присуджена Нобелівська премія за розробку теорії і практики оптимізації портфеля фондових активів.

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

Етапи ризик-менеджменту

У ризик-менеджменті прийнято виділяти декілька ключових етапів:

- на першому етапі відбувається виявлення ризику з супутньою оцінкою вірогідності його реалізації і масштабу наслідків;
- на другому етапі здійснюється розробка ризик-стратегії з метою зниження вірогідності реалізації ризику і мінімізації можливих негативних наслідків;
- на третьому етапі вибираються методи і інструменти управління виявленим ризиком;
- на четвертому етапі проводиться безпосереднє управління ризиком;
- на завершальному етапі оцінюються досягнуті результати і коректується ризик-стратегія.

За ключовий етап ризик-менеджменту вважається етап вибору методів і інструментів управління ризиком.

Методи і інструментарій ризик-менеджменту

Базовими методами ризик-менеджменту є відмова від ризиків, зниження, передача і ухвалення.

Ризик-інструментарій значно ширший. Він включає політичні, організаційні, правові, економічні, соціальні інструменти, причому ризик-менеджмент як система допускає можливість одночасного застосування декількох методів і інструментів ризик-управління.

Найбільш часто вживаним інструментом ризик-менеджменту є страхування. Страхування припускає передачу відповідальності за відшкодування передбачуваного збитку сторонній організації (страхової компанії).

Прикладами інших інструментів можуть бути відмова від надмірно ризикової діяльності (метод відмови), профілактика або диверсифікація (метод зниження), аутсорсинг витратних ризикових функцій (метод передачі), формування резервів або запасів (метод ухвалення).

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

Управління вимогами це процес запису, аналізу, трасування, пріоритезації і узгодження вимог та контролю змін і доведення до їх зацікавлених сторін. Це безперервний процес протягом всього життя проекту. Вимога – якість, якій мають відповідати результати проекту (продукту або послуги).

Мета управління вимогами полягає в тому, щоб переконатися, що організація відповідає потребам і очікуванням своїх клієнтів, внутрішніх або зовнішніх зацікавлених сторін. Управління вимогами починається з аналізу і виявлення цілей і обмежень організації. Управління вимогами додатково включає в себе підтримку планування вимог, інтеграції вимог і організації роботи з ними (атрибути для вимог).

Управління вимогами передбачає спілкування між членами проектної групи і зацікавленими сторонами, і адаптацію до змін у вимогах протягом всього проекту. Щоб запобігти перетину поля одного класу вимог з іншим, постійні зв'язки між членами команди розробників є критичними. Наприклад, при розробці програмного забезпечення для внутрішнього використання у бізнесу можуть бути настільки сильні потреби, що він може проігнорувати вимоги користувачів, або вважати, що створені сценарії використання покриють також і користувальницькі вимоги.

Відслідковування вимоги фактично означає документування всього життєвого циклу вимоги. Часто необхідно дізнатися першоджерело кожної вимоги. Для цього всі зміни вимог повинні бути задокументовані, щоб досягти стану повного відстеження. Відстежувати треба бути навіть використання реалізованих вимог.

Вимоги мають різні джерела, такі як ділова людина, що замовляє продукт, менеджер зі збуту і фактичний користувач. У всіх цих людей є різні вимоги до продукту. Використовуючи відслідковування вимог, реалізована в системі функція може бути простежена назад до людини або групі, яка замовляла її під час збору вимог. Ця особливість може, наприклад, використовуватися в процесі

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

розробки для пріоритезації вимог, визначаючи, наскільки цінною є дана вимога для певного користувача.

Відслідковування може також використовуватися після розгортання продукту. Наприклад, коли вивчення використання системи показує, що якась функція не використовується, можна визначити навіщо вона була потрібна спочатку.

Завдання управління вимогами

На кожному етапі процесу розробки існують ключові методи і задачі пов'язані з управлінням вимогами. Для ілюстрації, розглянемо наприклад стандартний процес розробки з п'ятьма фазами: дослідженням, аналізом здійсненності, дизайном, розробкою та тестуванням і випуском.

Дослідження. Під час фази дослідження збираються перші три класи вимог від користувачів, бізнесу і команди розробників. У кожній області задають однакові питання: які цілі, які обмеження, які використовуються процеси та інструменти і так далі. Тільки коли ці вимоги добре зрозумілі, можна приступати до розробки функціональних вимог.

Тут необхідне застереження: незалежно від того, як сильно група намагається це зробити, вимоги не можуть бути повністю визначені на початку проекту. Деякі вимоги змінюються, або тому що вони просто не були знайдені спочатку, або тому що внутрішні чи зовнішні сили торкаються проекту в середині циклу. Таким чином, учасники групи повинні спочатку погодитися, що головна умова успіху – гнучкість у мисленні та діях.

Результатом стадії дослідження є документ – специфікація вимог, схвалений усіма членами проекту. Пізніше, в процесі розробки, цей документ буде важливий для запобігання розповзанню меж проекту або непотрібних змін. Оскільки система розвивається, кожна нова функція відкриває світ нових можливостей, таким чином специфікація вимог прив'язує команду до оригінального бачення системи і дозволяє контрольоване обговорення змін.

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

У той час як багато організацій все ще використовують звичайні документи для керування вимогами, інші управляють своїми базовими вимогами, використовуючи програмні інструменти.

Ці інструменти керують вимогами використовуючи базу даних, і зазвичай мають функції автоматизації відстеження (наприклад, дозволяючи створювати зв'язки між батьківськими і дочірніми вимогами, або між тестами і вимогами), управління версіями, і управління змінами. Зазвичай такі інструментальні засоби містять функцію експорту, яка дозволяє створювати звичайний документ, екпортуючи дані вимог.

Аналіз здійсненності

На стадії аналізу здійсненності визначається вартість вимог. Для користувальницьких вимог поточна вартість роботи порівнюється з майбутньою вартістю встановленої системи. Задаються питання такі як: «Скільки нам зараз варті помилки введення даних?» Або, «Яка вартість втрати даних через помилки оператора пов'язаної з використанням інтерфейсом?». Фактично, потреба в новому інструменті часто розпізнається, коли подібні питання потрапляють до уваги людей, що займаються в організації фінансами.

Ділова вартість включає відповіді на такі питання як: «У якого відділу є бюджет на це?» «Який рівень повернення коштів від нового продукту на ринку?» «Який рівень скорочення внутрішніх витрат на навчання і підтримку, якщо ми зробимо нову, більш просту в використанні систему?»

Технічна вартість пов'язана з вартістю розробки програмного забезпечення та апаратною вартістю. «Чи є у нас потрібні люди, щоб створити інструмент?» «Чи потребуємо ми нове устаткування для підтримки нової системи?»

Подібні питання дуже важливі. Група повинна з'ясувати, чи буде новий автоматизований інструмент мати достатню ефективність аби перенести частину тягара користувачів на систему і зекономити час людей.

Ці питання також вказують на основну суть управління вимогами. Людина і інструмент формують систему, і це розуміння особливо важливе, якщо

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

інструмент – комп'ютер або новий додаток на комп'ютері. Людський розум вкрай ефективний у паралельній обробці та інтерпретації тенденцій з недостатніми даними. Комп'ютерний процесор ефективний у послідовній обробці і точному математичному обчисленні. Основна мета управління вимогами для програмного проекту полягала б у тому, щоб гарантувати, що автоматизована робота призначена «правильному» процесору.

Наприклад, «не змушуйте людину пам'ятати, де вона знаходиться в системі. Примусьте інтерфейс завжди повідомляти про місцезнаходження людини в системі». Або «не змушуйте людини вводити ті ж самі дані в два екрани. Примусьте систему зберігати дані і заповнювати їх де необхідно автоматично». Результатом стадії аналізу здійсненності є бюджет і графік проекту.

Дизайн. Припускаючи, що вартість точно визначена і переваги, які будуть отримані, є досить великими, проект може перейти до стадії проектування.

На стадії дизайну основна діяльність управління вимогами полягає в тому, щоб перевіряти чи відповідають результати дизайну документу вимог, щоб упевнитися, що робота залишається в межах проекту.

І знову, гнучкість є ключем до успіху. Ось класичний приклад змін проекту, які відмінно працювали. Проектувальники Форда на початку 1980-х очікували, що ціни на бензин піднімуться до 3,18 дол за галон до кінця десятиліття. На середині процесу дизайну автомобіля Ford Taurus, ціни встановилися приблизно на рівні 1,50 дол за галон. Колектив дизайнерів вирішив, що вони могли б створити більший, більш зручний, і більш потужний автомобіль, якщо б ціни на бензин залишилися низькими. Таким чином, вони перепроєктувати автомобіль. Коли новий автомобіль вийшов, він встановив загальнонаціональні рекорди продажів.

У більшості випадків, однак, відступ від оригінальних вимог до такої міри не працює. Таким чином документ вимог стає ключовим інструментом, який допомагає команді приймати рішення про зміни дизайну.

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

Розробка та тестування. На стадії розробки і тестування, основна діяльність управління вимогами – це гарантувати, що робота і ціна залишаються в межах графіка і бюджету, і що створюваний інструмент дійсно відповідає вимогам. Основним інструментом, використовуваним на цій стадії, є створення прототипу і ітераційне тестування. Для програмного додатка користувацький інтерфейс може бути створений на папері і перевірений з потенційними користувачами, в той час як створюється основа програми. Результати цих тестів записуються в керівництві по дизайну користувацького інтерфейсу і передаються колективу дизайнерів. Це економить їх час і робить їх завдання набагато простіше.

Під час роботи над програмою також використовувались підходи RUP.

Rational Unified Process (RUP) є ітеративним процесом розробки програмного забезпечення створеним Rational Software – підрозділом IBM з 2003.

RUP не є єдиним, конкретним розпорядчим процесом, а скоріше фреймворком процесу, що має бути адаптованим організаціями які займаються розробкою та командами розробників які оберуть елементи процесу, які підходять під їх потреби.

RUP являє собою продукт, спочатку розроблений Rational Software, яка була придбана компанією IBM в лютому 2003 року. Продукт містить у собі базу знань з гіперпосиланнями, та прикладами артефактів і докладні описи для різних видів діяльності. RUP входить в продукт IBM Rational Method Composer (RMC), який дозволяє налаштування процесу.

До 1997 року, Rational придбав Verdix, Objectory, Requisite, SQA, Performance Awareness, та Pure-Atria. Поєднання баз досвіду цих компаній привело до вироблення семи «найкращих практик» сучасної програмної інженерії:

1. Розробляти ітеративно, керуючись ризиками.
2. Управляти вимогами.
3. Використовувати компонентну архітектуру.

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

4. Моделювати програмне забезпечення візуально.
5. Постійно перевіряти якість.
6. Контролювати зміни.
7. Підлаштовуватись.

Ці найкращі практики рухали розробку продуктів Rational, та використовувались польовими командами Rational, щоб допомогти клієнтам вдосконалити якість, та передбачуваність їх розробницьких зусиль. Щоб зробити ці знання доступнішими, Філіпу Крачтену, було поставлено завдання збирати явні фреймворки сучасної розробки програмного забезпечення. Ці зусилля використовував заснований на HTML механізм доставки процесів розроблений Objectory.

У результаті «Раціональний уніфікований процес» (RUP) завершив стратегічну опору для Rational:

- Адаптовний процес що направляє розробку
- Інструменти, що автоматизують використання цього процесу
- Сервіси, що прискорюють впровадження і процесу, і інструментів.

Будівельні блоки RUP

RUP заснований на наборі будівельних блоків, чи містить елементи, що описують те, що повинно бути зробленим, необхідні навички, та покрокове пояснення того, як досягаються конкретні цілі розробки. Основними будівельними блоками, чи елементами вмісту, є наступні:

- Ролі (хто). Роль визначає набір навичок, компетенції та відповідальності.
- Робочі продукти (що). Робочий продукт являє собою щось отримане з завдання, в тому числі всі документи і моделі, випущені під час роботи впродовж процесу.
- Завдання (як). Завдання описує одиницю роботи, яка доручена ролі, яка забезпечує значущий результат.

У кожній ітерації, завдання діляться на дев'ять дисциплін: шість «інженерних дисциплін» (бізнес–моделювання, вимоги, аналіз і проектування,

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

реалізація, тестування, розгортання) і трьох допоміжних дисциплін (конфігурація і керування змінами, управління проектами, середовища).

Чотири фази життєвого циклу проекту

RUP визначає життєвий цикл проекту, що складається з чотирьох фаз. Ці фази дозволяють процесу, бути представленим на високому рівні, подібно до того як представляються проекти у «водоспадному» стилі, хоча, по суті, ключем до процесу є ітерації розробки, які простягаються вздовж всіх фаз. Крім того, кожен етап має одну ключову ціль, та віху в кінці, яка позначає досягнення цілі.

Початкова фаза

Первинною ціллю є адекватна оцінка системи, як база для обчислення початкових розцінок та бюджету. На цьому етапі встановлюються бізнес випадки, які включають бізнес-контекст, фактори успіху (очікувані доходи, визнання на ринку, і т.д.), а також фінансовий прогноз. На додаток до бізнес випадку генерується базова модель прецедентів, план проекту, попередня оцінка ризику і опис проекту (основні вимоги до проекту, обмеження та основні характеристики).

Після їх завершення проект перевіряється на відповідність наступним критеріям:

- Зацікавленими сторонами досягають згоди з визначення масштабів і оцінкою вартості/термінів.
- Розуміння вимог як свідчення якості первинних прецедентів.
- Достовірність оцінок вартості/термінів, пріоритетів, ризиків, та процесу розробки
- Глибина і ширина будь-якого архітектурного прототипу, який був розроблений.
- Встановлення базової лінії за допомогою якої можна порівняти фактичні витрати в порівнянні із запланованим витратам.

Якщо проект не пройде цей етап, що називається віхою життєвого циклу, він може бути як скасований так і повторений після реконструювання з метою кращого задоволення критеріїв.

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

Фаза уточнення

Основна мета полягає в пом'якшенні ключових ризиків, виявлених на основі аналізу до кінця цієї фази. Фаза уточнення – фаза де проект починає набувати форми. На цьому етапі робиться аналіз предметної області, і архітектура проекту отримує свою базову форму.

Ця фаза має пройти віху життєвого циклу архітектури (LCA), задовольняючи такі критерії:

– Модель прецедентів, в якій ідентифікуються прецеденти та актори, та розробляється більшість описів прецедентів. Модель прецедентів повинна бути завершена на 80%.

– Опис архітектури програмного забезпечення в процесі розробки програмної системи.

– Виконувана архітектура, яка реалізує архітектурно значимі прецеденти.

– Бізнес це випадки та список ризиків що переглядаються.

– План розвитку проекту в цілому.

– Прототипи, що явно зменшили кожен виявлений технічний ризик.

Якщо проект не може переступити цю віху, ще є час для того, щоб він був скасований або змінений. Тим не менше, після закінчення цього етапу, проект переходить в операцію з високим ступенем ризику, де зміни набагато складніші та згубні, при здійсненні. Системна архітектура є ключовим елементом розробки, що отримується з аналізу предметної області.

Фаза конструювання

Основна мета полягає в створенні програмної системи. На цьому етапі основна увага приділяється розробці компонентів та інших характеристик системи. Це етап, коли відбувається основна частина кодування. У більш великих проектах, може бути кілька фаз конструювання, в спробі поділити прецеденти на керовані сегменти, які можуть утворити презентабельні прототипи.

Цей етап створює перший реліз програмного забезпечення. Його завершення позначає віха початкової боєготовності.

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

Фаза впровадження

Основна мета полягає в переведенні системи з розробки у продукт, зробивши її доступною та зрозумілою для кінцевого споживача. Діяльність у рамках цієї фази включає навчання кінцевих користувачів та обслуговуючого персоналу, бета-тестування системи для перевірки її на відповідність очікуванням користувачів. Продукт також перевіряється на відповідність рівню якості, встановленого в початковій фазі.

Якщо всі вимоги задоволені, досягається віха релізу продукту, і цикл розробки завершується.

Шість інженерних дисциплін

Дисципліни бізнес-моделювання

Бізнес-моделювання пояснює, як описати бачення організації, в якій буде розгортатись система і як використати це бачення для виділення процесу, ролей та обов'язків.

Організації стають все залежнішими від ІТ систем, що вимагає від інженерів інформаційних систем знання того, як застосунок що вони розробляють вписується в організацію.

Підприємства інвестують в ІТ, коли вони розуміють, конкурентні переваги і вартість що додає технологія. Метою бізнес-моделювання є по-перше встановити глибше розуміння та комунікаційний канал між бізнес інженерією та програмною інженерією. Розуміння бізнесу означає, що програмісти повинні розуміти структуру і динаміку цільової організації (клієнта), нинішні проблеми в організації, а також можливі удосконалення.

Вони повинні також забезпечити загальне розуміння цільової організації між клієнтами, кінцевими користувачами та розробниками.

Дисципліни вимог

Вимоги пояснюють, як виявити запити зацікавлених осіб і перетворити їх в набір вимог, робочих продуктів, що досягають створювану систему й надають детальні вимоги до того, що система повинна робити.

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

Дисципліна аналізу та проектування

Метою аналізу і проектування, є показати, яким чином система буде реалізована. Ціллю є створення системи, яка:

- Виконує – в особливому середовищі реалізації – задачі та функції описані в описах прецедентів.
- Виконує всі свої вимоги.
- Легко змінити, коли змінюються функціональні вимоги.

Проектування дає в результаті модель проектування, а аналіз відповідно модель аналізу. Модель дизайну служить абстракцією вихідного коду; тобто модель дизайну працює «синьою», розміткою того як буде структурований та написаний вихідний код. Дизайн моделі складається проектування класів структурованих в пакети і підсистеми з чітко визначеними інтерфейсами, які представляють, що стане компонентами у реалізації. Він також містить опис того, як об'єкти цих сконструйованих класів співпрацюють для виконання прецедентів.

4.2 Захист розробленого програмного забезпечення

Розроблене програмне забезпечення захистимо за допомогою наступного алгоритму теорії кодування DSA.

Відправник і одержувач електронного документа використовують при обчисленні великі цілі числа: G і P – прості числа, L біт кожне ($512 < L < 1024$); q – просте число довжиною 160 біт (дільник числа $(P-1)$). Числа G , P , q є відкритими й можуть бути загальними для всіх користувачів мережі.

Відправник вибирає випадкове ціле число X , $1 < X < q$. Число X є секретним ключем відправника для формування електронного цифрового підпису. Потім відправник обчислює значення $Y = G^X \text{ mod } P$. Число Y є відкритим ключем для перевірки підпису відправника. Число Y передається всім одержувачам документів.

Цей алгоритм також передбачає використання однобічної функції

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

гешування $h(-)$. У стандарті DSS визначений алгоритм безпечного гешування SHA. Для того щоб підписати документ M , відправник гешує його в ціле геш-значення m : $m = h(M)$, $1 < m < q$, потім генерує випадкове ціле число K , $1 < K < q$, і обчислює число r : $r = (G^K \bmod P) \bmod q$. Потім відправник обчислює за допомогою секретного ключа X ціле число s :

$$s = \frac{m + r * X}{K} \bmod q .$$

Пара чисел r і s утворить цифровий підпис $S = (r, s)$ під документом M . Таким чином, підписане повідомлення являє собою трійку чисел $[M, r, s]$. Одержувач підписаного повідомлення $[M, r, s]$ перевіряє виконання умов $0 < r < q$, $0 < s < q$ і відкидає підпис, якщо хоча б одна із цих умов не виконана. Потім одержувач обчислює значення $w = 1/s \bmod q$, геш-значення $m = h(M)$ і числа $u_1 = (m * w) \bmod q$, $u_2 = (r * w) \bmod q$. Далі одержувач за допомогою відкритого ключа Y обчислює значення $v = ((G^{u_1} * Y^{u_2}) \bmod P) \bmod q$ і перевіряє виконання умови $v = r$. Якщо умова $v = r$ виконується, тоді підпис $S = (r, s)$ під документом M визнається одержувачем справжнім.

КБПЗ - 2024

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розглянемо розроблене ПЗ KVM системи на основі технології HDBase-T яке зображено на рисунку 5.1. Обчислення виконується через консольний інструмент з подальшою передачею результатів до інтерфейсу. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи:

– Навігаційне меню: Файл; KVM відеопотік; Шаблон; Налаштування; Довідка.

– Розділу інформації.

– Розділу виведення результату роботи системи.

– Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші.

– Функціональних кнопок ПЗ.

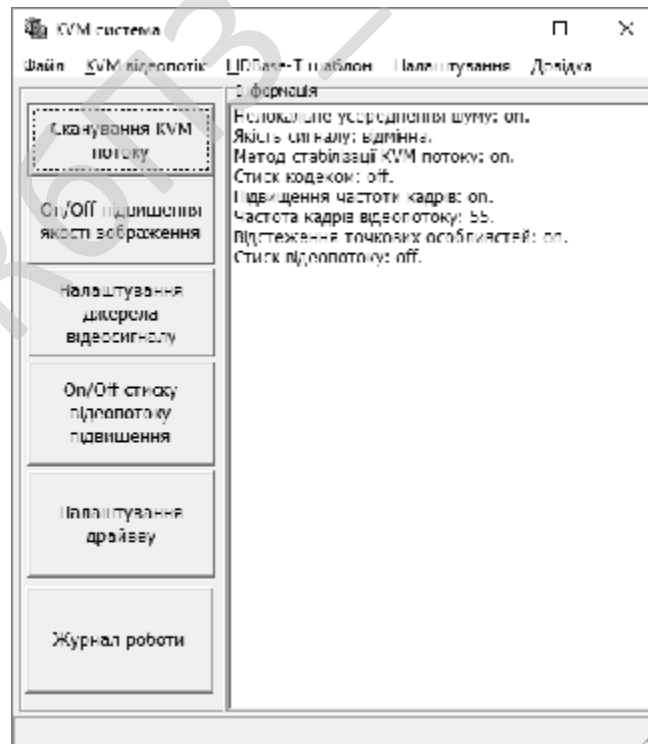


Рисунок 5.1 – Головне вікно ПЗ

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

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

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

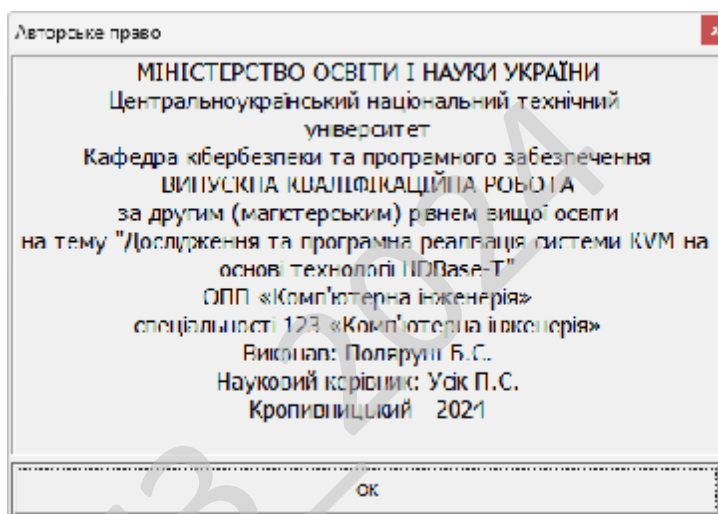


Рисунок 5.2 – Авторське право

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

Загалом процес розгортання складається з кількох взаємопов'язаних дій із можливими переходами між ними. Ця активність може відбуватися як з боку виробника так і з боку споживача. Оскільки кожна програмна система є

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

унікальною, то усі процеси та процедури під час розгортання важко передбачити. Тому, "розгортання" можна трактувати як загальний процес відповідно до певних вимог та характеристик. Розгортання може здійснюватись програмістом і в процесі розробки програмного забезпечення.

До діяльностей пов'язаних із розгортанням програмного забезпечення відносять:

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

При впровадженні програмного забезпечення потрібно урахувати наступні дії:

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

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

– Виконання робіт з загальної стандартизації існуючої діяльності організації, коли виділяються кращі практики виконання процедур і включаються

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

в IT рішення за принципом найбільшої корисності для більшості учасників. Відсоток таких процедур щодо загального обсягу автоматизації може бути невеликий, але це надає процесу побудови рішення вагу в організації за рахунок збільшення його необхідності.

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

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

Проводилась оцінка:

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

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

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

Проводилось тестування форматом білої скриньки засноване на аналізі керуючої структури програми. Програма вважається повністю перевіреною, якщо проведено вичерпне тестування маршрутів (шляхів) її графа управління.

У цьому випадку формуються тестові варіанти, в яких:

- Гарантується перевірка всіх незалежних маршрутів програми.
- Знаходяться гілки True, False для всіх логічних рішень.
- Виконуються всі цикли (у межах їхніх кордонів та діапазонів).
- Аналізується правильність внутрішніх структур даних.

Недоліки тестування "білої скриньки":

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

- Кількість незалежних маршрутів може бути дуже велика.
- Повне тестування маршрутів не гарантує відповідності програми вихідним вимогам до неї.
- У програмі можуть бути пропущені деякі маршрути.
- Не можна виявити помилки, поява яких залежить від даних.

Переваги тестування "білої скриньки" пов'язані з тим, що принцип «білої скриньки» дозволяє врахувати особливості програмних помилок:

- Кількість помилок мінімально в «центрі» і максимально на «периферії» програми.
- Попередні припущення про ймовірність потоку керування або даних у програмі часто бувають некоректними. У результаті типовим може стати маршрут, модель обчислень за яким опрацьована слабо.

– При записі алгоритму програмного забезпечення у вигляді тексту на мові програмування можливе внесення типових помилок трансляції (синтаксичних та семантичних).

– Деякі результати в програмі залежать не від вихідних даних, а від внутрішніх станів програми.

Обрано умови розповсюдження – proprietary software.

Програмне забезпечення, на яке зберігаються як немайнові, так і майнові авторські права. Отримавши або придбавши таке програмне забезпечення, користувач отримує обмежені права користування ним: може бути заборонено або закрито доступ до коду (вивчення), внесення змін, тиражування, розповсюдження та перепродаж. Програмне забезпечення вважається власницьким, якщо наявне хоча б одне з перелічених обмежень.

Найчастіше основним методом захисту майнових прав на власницьке ПЗ, поза ліцензійною угодою, власник обирає закриття сирцевого коду, захищаючи свій продукт від модифікації і вбудовуючи системи обмеження користування через авторизацію. Таке програмне забезпечення називається закритим. Проте,

код власницького продукту може бути і відкритим, але власник може обмежити права користувача умовами користувацької ліцензії.

Власницьке програмне забезпечення та комерційне програмне забезпечення не є синонімами – власницьким може бути і безплатне (тобто, некомерційне) програмне забезпечення.

На протипагу власницькому ПЗ існує вільне програмне забезпечення, автори і власники якого дозволяють вивчати, модифікувати і поширювати свій продукт. Саме визначення власницького програмного забезпечення виникло в результаті діяльності громадського руху вільного програмного забезпечення (представленого Фондом вільного програмного забезпечення та іншими організаціями) і осмислення умов свободи користування програмами. Визначенням власницького програмного забезпечення є не невідповідність хоча б одній з базових умов вільного програмного забезпечення. Сама назва власницьке ПЗ підкреслює визначальне значення власника у способі використання і можливостях розвитку цього програмного забезпечення.

КБПЗ – 2024

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи KVM на основі технології HDBase-T.

Метою розробки є дослідження та програмна реалізація системи KVM на основі технології HDBase-T.

Об'єктом дослідження є процес KVM на основі технології HDBase-T.

Предметом дослідження є методи KVM на основі технології HDBase-T.

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

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

- Удосконалено метод KVM на основі технології HDBase-T.
- Розроблено вітчизняний продукт KVM на основі технології HDBase-T, який має більш широкі можливості, на відміну від існуючих аналогів.

					VKPM-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

7 МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ІТ-ПРОЄКТУ

7.1 Визначення цільової аудиторії кінцевого готового продукту

Результати дослідження та програмної реалізації системи KVM (Keyboard, Video, Mouse) на основі технології HDBase-T можуть зацікавити аудиторію представлену на рисунку 7.1.

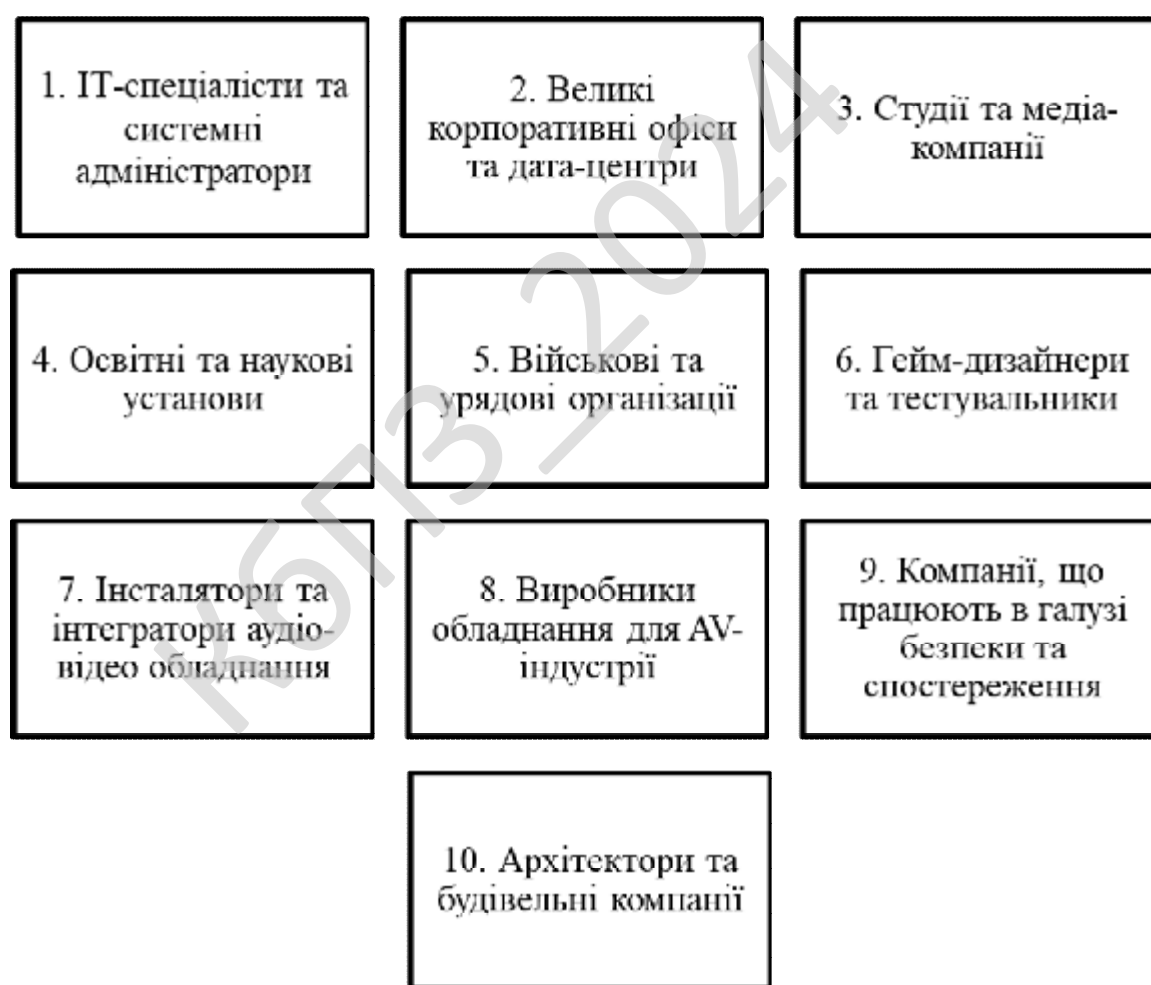


Рисунок 7.1 – Цільова аудиторія

Більш розширено про цільову аудиторію можемо сказати наступне.

ІТ-спеціалісти та системні адміністратори зацікавлені в управлінні віддаленими серверами, мережевими пристроями та іншими критичними ІТ-ресурсами. Використання KVM на основі HDBase-T може забезпечити стабільний доступ до обладнання з високою роздільною здатністю відео та мінімальними затримками.

Великі корпоративні офіси та дата-центри потребують централізованого управління великою кількістю комп'ютерів та серверів, зокрема для оптимізації простору та зменшення витрат на інфраструктуру. HDBase-T забезпечує можливість передачі відео, аудіо, управління і даних через один кабель на великі відстані, що є критично важливим для масштабних об'єктів.

Студії та медіа-компанії потребують високоякісного відеоз'єднання та надійного управління пристроями в умовах прямих ефірів або монтажу відео. HDBase-T дозволяє передавати відео в форматі 4K без компресії, що є важливим для професійної медіа-індустрії.

Освітні та наукові установи використовують KVM-системи для управління навчальними та дослідницькими лабораторіями, де є потреба контролювати декілька комп'ютерів з одного місця. Можливість підключення великих дисплеїв або проекторів для демонстрацій та навчання з мінімальними затримками.

Військові та урядові організації використовують KVM-системи для управління віддаленими об'єктами з підвищеними вимогами до безпеки та стабільності зв'язку. HDBase-T технологія може бути корисною для роботи в складних умовах, де важлива надійність з'єднання.

Гейм-дизайнери та тестувальники зацікавлені в управлінні тестовими стендами з потужними комп'ютерами та консолями для розробки і тестування ігор. HDBase-T забезпечує високоякісне зображення та стабільний контроль навіть при передачі на великі відстані, що важливо для тестування ігрового обладнання.

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

Інсталятори та інтегратори аудіо-відео обладнання встановлюють та налаштовують професійні AV-системи в корпоративних, освітніх чи промислових середовищах. Зацікавлені у використанні технології HDBase-T для створення надійних і масштабованих AV-рішень з простим управлінням.

Виробники обладнання для AV-індустрії потребують досліджень та готових рішень на основі HDBase-T для інтеграції в свої продукти, зацікавлені у можливості розширити функціонал своїх рішень для професійного ринку відеотрансляцій та управління.

Компанії, що працюють в галузі безпеки та спостереження використовують KVM-системи для моніторингу та управління камерами спостереження, системами сигналізації та іншим обладнанням безпеки. HDBase-T може забезпечити стабільне та надійне з'єднання для передачі відео з камер високої роздільної здатності.

Архітектори та будівельні компанії потребують інтеграції KVM-систем на етапі планування інфраструктури великих об'єктів, таких як офіси, конференц-зали або навчальні установи. HDBase-T технологія дозволяє спрощувати кабельне з'єднання, що важливо при плануванні сучасних технологічних просторів.

Ці категорії мають специфічні потреби, які технологія KVM на базі HDBase-T може ефективно задовольнити, підвищуючи продуктивність та зменшуючи витрати на обслуговування.

7.2 Оцінка привабливості шляхом застосування методів експертних оцінок

Оцінка привабливості програмної реалізації системи KVM на основі технології HDBase-T може бути здійснена за допомогою експертних оцінок. Це включає збір думок фахівців у відповідних галузях для визначення переваг і недоліків системи, а також потенційних можливостей на ринку.

На першому етапі потрібно визначити критерії, за якими буде оцінюватися система (рисунок 7.2).

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

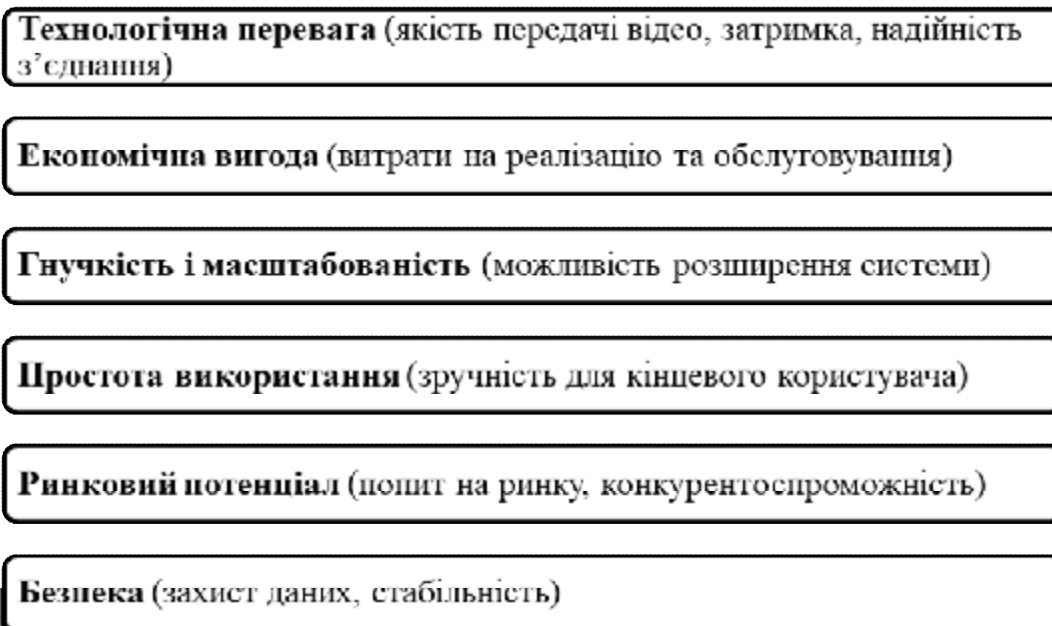


Рисунок 7.2 – Критерії експертної оцінки

Далі залучаємо експертів із різних галузей, таких як: інженери з ІТ та телекомунікацій, системні адміністратори, спеціалісти з аудіо-відео технологій, маркетологи, користувачі, що мають досвід роботи з подібними системами.

Експерти оцінюють кожен критерій за шкалою, наприклад, від 1 до 5, де 1 – дуже погано, а 5 – відмінно. Далі зводимо усереднене значення критеріїв за оцінками експертів. Результати розрахунку зводимо в таблицю 7.1.

Таблиця 7.1 – Зведені результати експертних оцінок

Критерій	Оцінка експертів
Технологічна перевага	4
Економічна вигода	3
Гнучкість і масштабованість	4
Простота використання	5
Ринковий потенціал	4
Безпека	4

На основі отриманих оцінок експертів можна зробити висновки про привабливість реалізації системи KVM на базі HDBase-T:

Сильні сторони: висока оцінка за простоту використання та технологічну перевагу вказує на конкурентоспроможність системи.

Слабкі сторони: середня оцінка економічної вигоди свідчить про необхідність подальшого аналізу витрат та можливостей зниження ціни.

Рекомендації: Слід зосередитися на маркетингових кампаніях, які підкреслюють переваги системи, та розробити стратегії для зниження витрат на реалізацію.

Цей підхід дозволяє системно оцінити привабливість програмної реалізації системи KVM на основі HDBase-T і підкреслити її потенційні можливості та ризики.

7.3 Вибір методу оцінки вартості ПЗ

Оцінка вартості програмної реалізації системи KVM на основі технології HDBase-T може бути здійснена за допомогою кількох методів, але найбільш підходящими є: метод аналізу витрат, метод порівняльного аналізу, метод оцінки на основі вартості, метод оцінки з урахуванням ризиків, метод аналізу інвестиційної привабливості. Залежно від доступності інформації, можна комбінувати кілька методів. Метод аналізу витрат може бути основним, а порівняльний аналіз може допомогти в уточненні розрахунків. Також варто розглянути метод оцінки на основі вартості, щоб забезпечити максимальну вигоду від реалізації системи. Важливо також залучити експертів для оцінки та верифікації отриманих результатів, оскільки це дозволить знизити ризики і підвищити точність оцінки вартості.

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

Економічна ефективність від впровадження системи KVM (Keyboard, Video, Mouse) на основі технології HDBase-T може бути проаналізована за

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

допомогою кількох ключових показників, що демонструють потенційні вигоди для компанії або організації. Нижче наведено вигоду та ключову проблему, що вирішується для клієнта (рис. 7.3.).



Рисунок 7.3 – Економічна ефективність від реалізації проєкту для клієнта

Впровадження системи KVM на основі HDBase-T може забезпечити значну економічну ефективність для організації, зменшуючи витрати на обладнання, електроенергію, обслуговування та підвищуючи продуктивність працівників. Це ілюструє як система може стати вигідним інвестиційним проєктом.

7.5 Пропозиція алгоритму просування проекту розробки ПЗ

Ось запропонований алгоритм просування проекту програмної реалізації системи KVM на основі технології HDBase-T. Цей алгоритм охоплює етапи від дослідження ринку до запуску і підтримки продукту (рисунок 7.4).

Цей алгоритм просування проекту програмної реалізації системи KVM на основі технології HDBase-T допоможе організувати ефективний процес від дослідження ринку до підтримки клієнтів. Основна мета полягає в тому, щоб максимізувати видимість продукту, залучити нових клієнтів і забезпечити високий рівень задоволеності споживачів.

7.6 Оптимізація каналів збуту та шляхів реалізації ПЗ

Оптимізація каналів збуту та шляхів реалізації проекту програмної реалізації системи KVM на основі технології HDBase-T є важливим етапом для забезпечення успішної комерційної діяльності. Ось кілька стратегій та рекомендацій для цього:

- мультимодальний підхід до каналів збуту;
- електронна комерція;
- цільова реклама та маркетинг;
- вдосконалення логістики;
- постпродажне обслуговування;
- моніторинг і аналіз;
- використання нових технологій.

Оптимізація каналів збуту та шляхів реалізації системи KVM на основі HDBase-T вимагає комплексного підходу, що поєднує в собі різні стратегії продажу, маркетингу та підтримки клієнтів. Важливо постійно адаптуватися до змін у ринкових умовах та потребах споживачів для досягнення максимального успіху.

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

1. Дослідження ринку

- **Аналіз конкурентів:** Вивчення ринку, виявлення основних конкурентів, їхніх продуктів і стратегій.
- **Визначення цільової аудиторії:** Ідентифікація потенційних користувачів, їхніх потреб і вподобань.
- **Оцінка попиту:** Визначення потреб у системах KVM на основі HDBase-T, аналіз тенденцій ринку.

2. Розробка стратегії просування

- **Формування Унікальної Пропозиції Продажу (USP):** Визначення основних переваг продукту, які відрізняють його від конкурентів.
- **Вибір каналів комунікації:** Визначення каналів для просування (онлайн, офлайн, B2B, B2C).
- **Створення контент-плану:** Розробка контенту для соціальних мереж, веб-сайту, блогів, e-mail розсилок.

3. Розробка та тестування продукту

- **Розробка MVP (Minimum Viable Product):** Створення базової версії продукту для тестування ринку.
- **Тестування продукту:** Залучення фокус-групи або бета-тестувальників для збору зворотного зв'язку та покращення продукту.

4. Запуск кампанії з просування

- **Старт рекламної кампанії:** Використання цифрових каналів (SEO, контекстна реклама, соціальні мережі) для залучення трафіку.
- **Організація вебінарів та демонстрацій:** Проведення онлайн-презентацій продукту, де потенційні клієнти можуть ознайомитися з його функціональністю.
- **Виставки та конференції:** Участь у галузевих заходах для демонстрації продукту та налагодження контактів.

5. Моніторинг та аналіз результатів

- **Відстеження KPI:** Аналіз ключових показників ефективності (кількість продажів, відвідуваність сайту, конверсії).
- **Збір зворотного зв'язку:** Опитування клієнтів та фокус-груп для оцінки задоволеності продуктом і сервісом.
- **Аналіз конкурентного середовища:** Постійний моніторинг дій конкурентів та змін у ринкових умовах.

6. Корекція стратегії

- **Внесення змін на основі зворотного зв'язку:** Внесення коректив у продукт або маркетингову стратегію на основі відгуків і результатів кампанії.
- **Оптимізація каналів просування:** Визначення найбільш ефективних каналів і їх подальше використання.

7. Підтримка та обслуговування клієнтів

- **Запровадження служби підтримки:** Створення технічної підтримки для клієнтів, забезпечення навчальних матеріалів.
- **Постійне вдосконалення продукту:** Оновлення системи на основі нових технологій та потреб ринку.

Рисунок 7.4 – Алгоритм просування проекту

7.7 Визначення ключових факторів успіху конкретного проєкту

Ключові фактори успіху проєкту програмної реалізації системи KVM на основі технології HDBase-T можуть варіюватися в залежності від специфіки ринку та цілей проєкту (рис. 7.5).

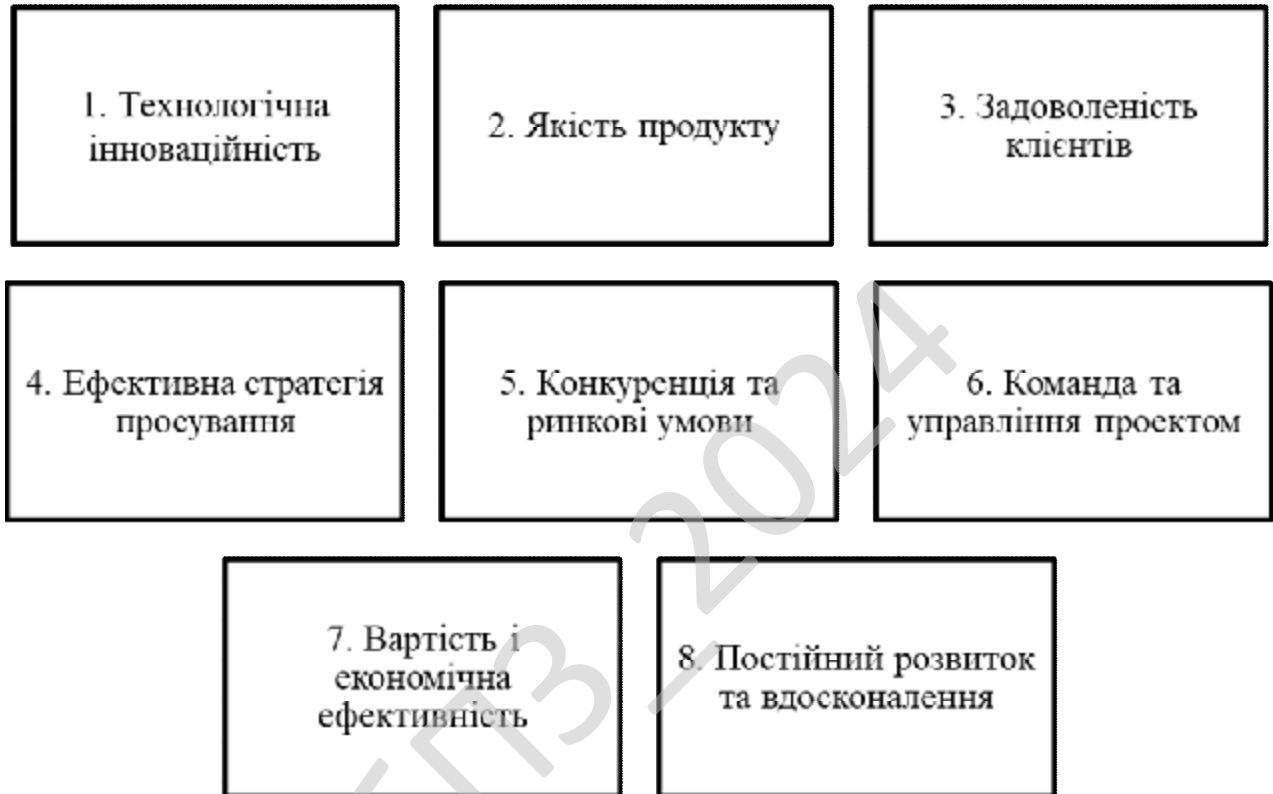


Рисунок 7.5 – Ключові фактори успіху проєкту

Успіх проєкту програмної реалізації системи KVM на основі технології HDBase-T залежить від багатьох факторів, включаючи технологічні аспекти, якість продукту, задоволеність клієнтів, ефективність маркетингових стратегій, команду управління проєктом та фінансову стратегію. Постійний моніторинг та адаптація до змін у ринкових умовах допоможуть досягти успіху в цьому конкурентному середовищі.

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Законом України “Про охорону праці” [3] регламентуються загальні положення державної політики в галузі охорони праці, а реалізуються ці положення нормативно-правовими актами про охорону праці, зокрема Наказом Міністерства соціальної політики України № 207 від 25.04.2018 р. та №508/31960 «Про затвердження Вимог щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями» [5], яким затверджено нормативно-правовий акт з охорони праці НПАОП 0.00-7.15-18, «Правила охорони праці під час експлуатації електронно-обчислювальних машин», та «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98.

Програмісти у процесі роботи мають негативний вплив на органи зору, а також мають значну розумову напругою і нервово-емоційне навантаження. Руки (суглоби пальців та м’язи рук) при роботі з клавіатурою мають теж істотне навантаження. До шкідливих факторів, які впливають на робітників галузі інформаційних технологій (ІТ) спеціалісти відносять високочастотні електромагнітні коливання (випромінювання) роботи апаратної частини ЕОМ та виділення шкідливих газів.

Ці шкідливі фактори можуть привести до професійних захворювань.

Керуючись наступними нормативно-правовими актами: «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98 [5], та «Вимоги щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями» НПАОП 0.00-7.15-18 розглянемо шкідливі чинники роботи персоналу.

Щоб запропонувати заходи щодо зменшення негативного впливу

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

комп'ютера на організм людини визначемо фактори, які можуть викликати професійне захворювання і впливають на працездатність програміста.

8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

Працівники, задіяні на роботах, пов'язаних з періодичною або постійною роботою за комп'ютером, піддаються впливу факторів виробничої небезпеки, основними з яких наступні:

1. Фізичні:

- Підвищений рівень напруги в електричному ланцюзі, замикання якої може пройти через тіло працюючого;
- Підвищений рівень рентгенівського випромінювання;
- Підвищений рівень ультрафіолетового випромінювання;
- Підвищений рівень інфрачервоного випромінювання;
- Можливість ураження статичною електрикою;
- Запиленість повітря робочого приміщення;
- Підвищений вміст важких (+) аероіонів;
- Нерівномірний розподіл яскравості в полі зору;
- Підвищений рівень пульсації світлового потоку;

2. Хімічні: підвищений вміст у повітрі вуглекислого газу, озону, аміаку, фенолу, формальдегіду та інше

3. Психофізіологічні:

- Напруга зору;
- Напруга пам'яті;
- Напруга уваги.;
- Тривале статичне напруження;
- Відносно великий обсяг інформації, що обробляється в одиницю часу;
- Монотонність праці в окремих випадках;
- Нераціональна організація робочого місця.

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

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

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

Мікроклімат повітряного середовища в приміщенні характеризується запиленістю та загазованістю повітря. Мікроклімат приміщення визначається діючим на організм людини поєднанням, вологості, температури, швидкості руху повітря та інтенсивності теплового випромінювання. Аналіз мікроклімату складається з визначення зазначених вище факторів і порівняння результатів із встановленими нормами.

У таблиці 8.3 наведено оптимальні та фактичні значення параметрів мікроклімату як для категорії ваги робіт Ia, так і розглянутого приміщення. У приміщеннях, де встановлено ЕОМ, рекомендується застосування тільки оптимальних значень показників мікроклімату.

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

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

В літню пору застосовується кондиціонер.

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

Таблиця 8.3 – Оптимальні і фактичні значення параметрів мікроклімату

Пора року	Оптимальні для Ia			Фактичні		
	Температура, °C	Воло- гість,%	Швидкість повітря, м/с	Температура, °C	Воло- гість%	Швидкість повітря, м/с
Холодна	22-24	40-60	0,1	21-22	46-54	0,1
Тепла	23-25	50-70	0,1	24-25	50-65	0,11

У приміщенні знаходяться наступні джерела шуму: принтер XEROX PHASER 3020BI, електродвигуни вентиляторів ЕОМ.

Одним з найважливіших факторів, які впливають на ефективність трудової діяльності людини, та попереджають травматизм і професійні захворювання програмістів є освітлення на робочому місці.

З 2019 року діють Державні будівельні норми України “Природне і штучне освітлення” – ДБН В.2.5-28:2018 [1], у яких прописані вимоги до використання всіх освітлювальних приладів, у т.ч. світлодіодних.

Працю працівника, який постійно працює за комп’ютером, згідно ДБН В.2.5-28:2018 [1], можна віднести до роботи з малою точністю (найменший розмір об’єкта розрізнення від 1 до 5 мм) V-го розряду зорової роботи, з великою контрастністю об’єкта розрізнення (символів на екрані дисплея), з темним тлом (під розряд зорової роботи B).

Приміщення можна віднести до 1-ої групи приміщень, у яких проводиться розрізнення об'єктів зорової роботи при фіксованому напрямку лінії зору того, що працює на робочу поверхню.

Для такого типу приміщень і розряду зорової роботи нормоване значення коефіцієнта природної освітленості (КПО) робочої поверхні (при поєднаному, спільному освітленні), повинен становити не більше 1,5%, освітленість при штучному висвітленні повинна становити 300 Лк. [1], Крім того все поле зору повинне бути освітлено достатньо рівномірно – ця основна гігієнічна вимога. Так як яскраве світло на ділянці периферійного зору значно збільшує напруженість очей і, як наслідок, призводить до їх швидкої стомлюваності, ступінь освітлення приміщення і яскравість екрану комп'ютера повинні бути приблизно однаковими.

8.4 Розробка заходів з умов поліпшення охорони праці

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

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

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

Рекомендовані заходи: регулярні періодичні наочні огляди персоналом шляхів для евакуації людей із приміщення, відповідно до плану евакуації (який повинен розташовуватись на видному місці у приміщенні), включення до колективного договору мінімально можливого вмісту аптечок з обов'язковою

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

наявністю масок-клапанів, або іншого спорядження для штучного дихання. Регулярна періодична перевірка параметрів заземлення та занулення (вимірювання опору ланцюга).

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

8.5 Розрахункова частина

Проведемо розрахунок штучного освітлення за методом коефіцієнта використання світлового потоку для приміщення ширина якого складає 4 м, довжина – 5 м, висота – 3 м.

Для того, щоб визначити потрібну кількість світильників, які повинні забезпечити нормований рівень освітленості, необхідно враховувати, що, з одного боку «середня освітленість робочих місць з постійним перебуванням людей повинна бути не менш як 200 люкс», а з іншого «штучне освітлення при системі комбінованого освітлення для зорової роботи найвищої точності повинна складати 300 люкс» та вимогу ДБН В.2.5-28:2018, що «створювати освітленість більше ніж 300 лк при світлодіодних світильниках дозволяється тільки за наявності обґрунтування».

Таким чином для розрахунку приймаємо середню освітленість робочих місць 300 Люкс [1].

Визначимо світловий потік, що падає на робочу поверхню за формулою [6]:

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

де: F – світловий потік, що розраховується, Лм;

E – нормована мінімальна освітленість, Лк; $E = 300$ Лк. [1];

S – площа поверхні, на яку падає світловий потік, м² [1];

Z – відношення середньої освітленості до мінімальної (зазвичай

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		80

приймається рівним 1.1... 1.2, в нашому випадку $Z = 1,1$);

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

n – коефіцієнт використання світлового потоку, (відношення світлового потоку, що падає на розрахункову поверхню, до сумарного потоку всіх ламп і обчислюється в долях одиниці; залежить від характеристик світильника, розмірів приміщення, забарвлення стін і стелі, що характеризуються коефіцієнтами відбиття від стін ($\rho_{стін.}$) і стелі ($\rho_{стелі}$), значення коефіцієнтів дорівнюють $\rho_{стін} = 50\%$ і $\rho_{стелі} = 50\%$ [6].

Обчислимо індекс приміщення за формулою:

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

де: S – площа поверхні, на яку падає світловий потік, m^2 [1];

h – розрахункова висота підвісу, $h = 3$ м;

A – ширина приміщення, $A = 4$ м;

B – довжина приміщення, $B = 5$ м.

Підставимо всі значення у формулу та визначимо індекса приміщення:
 $i = 0,74$.

Знаючи індекс приміщення, за знаходимо $n = 0,29$ (з табличних даних коефіцієнтів використання світлового потоку (n) світильників відповідного типу [6]). Підставимо всі значення у формулу, визначимо світловий потік: $F = 43043$ Лм.

Будемо використовувати світильники (світлодіодні панелі) Matt White Армстронг 36W 6000К, світловий потік яких $F_n = 3000$ Лм.

Число ламп визначається по формулі:

$$N = F / F_n$$

де: F – світловий потік,

F_n – світловий потік одного світильника.

Підставимо всі значення у формулу та визначимо індекса приміщення:

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

$$N = 43043 / 3000 = 14,34 \text{ шт.}$$

Приймаємо необхідну кількість світильників 15 шт.

Висновки до розділу

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

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

КБПЗ - 2024

					VKPM-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи KVM на основі технології HDBase-T.

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

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти наведені теоретичне узагальнення й рішення наукового завдання дослідження методів KVM на основі технології HDBase-T.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем KVM на основі технології HDBase-T.

– Досліджена система KVM на основі технології HDBase-T.

– На основі отриманих результатів досліджень створена програмна реалізація системи KVM на основі технології HDBase-T.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання KVM на основі технології HDBase-T.

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

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

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

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

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

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм DSA.

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

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

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Поляруш Б.С. Дослідження та програмна реалізація системи KVM на основі технології HDBase-T // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2024.
2. Henry Lloyd. Interactive Computer Graphics. States Academic Press. 2022. 247 p.
3. Ranjan Parekh. Fundamentals of Image, Audio, and Video Processing Using MATLAB® With Applications to Pattern Recognition. CRC Press. 2021. 406 p.
4. Alasdair McAndrew. A Computational Introduction to Digital Image Processing. Chapman & Hall. 2021. 560 p.
5. Peter Shirley, Steve Marschner. Fundamentals of Computer Graphics. 2009
6. Михайло Пічугін, Іван Канкін, Володимир Воротніков Комп'ютерна графіка. Навчальний посібник / Центр навчальної літератури 346 с. 2019р.
7. Маценко В.Г. Комп'ютерна графіка: Навчальний посібник. – Чернівці: Рута, 2009 – 343 с.
8. Інженерна комп'ютерна графіка: підручник / В.В. Проців [та ін.] / М-во освіти і науки України, Нац. гірн. унт-т. – Дніпро: НГУ, 2017. – 247 с.
9. Проців В.В. Прикладна комп'ютерна графіка [Текст]: Навч. посібник / В.В. Проців, К.А. Зіборов, К.М. Бас, Г.К. Ванжа; М-во освіти і наук, Нац. гірн. унт. – Д.: НГУ, 2016. – 187 с.
10. Kopf, Johannes and Lischinski, Dani. Depixelizing Pixel Art (англ.) // ACM Trans. Graph. – 2011. – Vol. 30, no. 4. – P. 99:1--99:8.
11. Giachetti, Andrea and Asuni, Nicola. Real-Time Artifact-Free Image Upscaling (англ.) // Trans. Img. Proc.. – 2011. – Vol. 20, no. 10. – P. 2760—2768.
12. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56.

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

14. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». *Lecture Notes on Data Engineering and Communications Technologies*, 2023, 178, pp. 208–223.

15. Smirnov, O., Neskrodieva, T., Fedorov, E., Rudakov, K., Neskrodieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» *CEUR Workshop Proceedings*, Volume 3187, 2022,

16. Smirnov O., Smirnova T., Anas M. Al-Oraiqat, Drieiev O., Polishchuk L., Sheroz Khan, Yassin M. Y. Hasan, Aladdein M. Amro, Hazim S. AlRawashdeh «Method for Determining Treated Metal Surface Quality Using Computer Vision Technology». *Sensors (Basel, Switzerland)* Volume 22, Issue 16, 6223, 2022.

17. Smirnov O., Kuznetsov A., Kryvinska N., Kiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». *SN Computer Science*, Vol 2, 337, 2021. <https://doi.org/10.1007/s42979-021-00739-w>

18. Smirnov O., Kuznetsov A., Zhora V., Onikiychuk A., Pieshkova O. «Hiding Messages in Audio Files Using Direct Spread Spectrum». 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2021, Cracow, Poland, 22-25 September 2021. P. 414-418.

19. Smirnov O., Kuznetsov A., Lokotkova I., Kuznetsova T., Florov S., Lebid O. «Using Orthogonal Signals to Hide Information in Images». 4 IEEE International Conference on Advanced Information and Communication Technologies (AICT) – 2021, Lviv, Ukraine, September 21-25, 2021. P. 255-260.

20. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256.

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

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

23. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». CEUR Workshop Proceedings Volume 2654, 2020, Pages 1-14.

24. Smirnov O., Kuznetsov A., Onikiychuk A., Makushenko T., Anisimova O., Arischenko A. «Adaptive pseudo-random sequence generation for spread spectrum image steganography». 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 161-165.

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

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

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

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

29. Smirnov, O., Kuznetsov, A., Gorbacheva, L., Babenko, V., «Hiding data in images using a pseudo-random sequence», CEUR Workshop Proceedings Volume 2608, 2020, Pages 646-660.

30. Zhurakovskiy, B., Tsopa, N., Batrak, Y., Odarchenko, R., Smirnova, T «Comparative analysis of modern formats of lossy audio compression». Workshop Proceedings, 2020, 2654, стр. 315-327.

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

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

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

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

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

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

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

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

39. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New Technique for Hiding Data in Cover Images Using Adaptively Generated Pseudorandom Sequences». CEUR Workshop Proceedings Volume 2732, 2020, Pages 214-227.

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

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

42. Т.В. Смірнова, О.М. Дреєв, О.А. Смірнов «Хмарна інформаційна система оцінювання шорсткості з використанням дискретного частотного аналізу макروفотografій». IV міжнародна науково-практична конференція «Інформаційна безпека та комп'ютерні технології», м. Кропивницький. 15-16 квітня 2021р. – Кропивницький: ЦНТУ. – 2021. – С. 30.

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

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

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

46. О. Смірнов, Є. Деменко, О. Онікійчук, А. Арищенко, Л. Горбачова, «Формування псевдовипадкових послідовностей для приховування даних в зображеннях» Комп'ютерні науки та кібербезпека. № 4. С. 30-37. 2019.

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

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

49. Смірнов О.А., Дреєва Г.М. Метод генерування фрактального трафіку за допомогою моделі генератора на графі. Монографія: Інформаційна

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

безпека та інформаційні технології : монографія / за заг. ред. В. С. Пономаренка. – Х. : Вид. Рожко С.Г. 2019. С. 123-139

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

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

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

53. Смірнов О.А., Кавун С.В., Коваленко О.В., Доренський О.П., Дреєв О.М., Вялкова В.І. Комп'ютерні мережі. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 233 с.

54. Смірнов О.А., Дреєв О.М. Порівняння бітових щільностей при використанні різних методів кодування інформації. Збірник наукових праць "Системи обробки інформації". – Випуск 2 (118). т.2. – Х.: ХУПС – 2014. – С. 64-67

55. Смірнов О.А., Дреєв О.М. Порівняння бітових щільностей при використанні різних методів кодування інформації. Збірник тез VI міжнародної науково-практичної конференції “Проблеми та перспективи розвитку ІТ-індустрії”. м. Харків. 17-18 квітня 2014р. – Харків: ХНЄУ. – 2014. – С. 240.

					ВКРМ-123.24.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

Додаток А
(обов'язковий)

Технічне завдання

Зміст

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

					ВКРМ-123.24.0032.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Поляруш Б.С.				<i>Дослідження та програмна реалізація системи KVM на основі технології HDBase-T</i>	Літ.	Аркуш	Аркушів
Перевірів	Усік П.С.					М	1	6
Н. Контр.	Коваленко А.С.				ЦНТУ КІ-23М			
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи KVM на основі технології HDBase-T.

2 Підстава для розробки

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

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи KVM на основі технології HDBase-T.

4 Джерела розробки

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

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

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

					ВКРМ-123.24.0032.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

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

5.2 Показники призначення

Система повинна забезпечувати:

- програмну реалізацію системи KVM на основі технології HDBase-T;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

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

5.4 Вимоги до архітектури

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

5.5 Вимоги до надійності

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

					ВКРМ-123.24.0032.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

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

5.7 Вимоги до складу та параметрів технічних засобів

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

5.8 Вимоги до інформаційної і програмної сумісності

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

5.8.1 Обладнання

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

5.8.2 Мова програмування

Середовище Python.

					ВКРМ-123.24.0032.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

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

7 Економічні вимоги

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

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

8 Вимоги щодо охорони праці

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

					ВКРМ-123.24.0032.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

9 Перелік документів, що розробляються

– Наукова новизна	– 1 аркуш.
– Структурна схема системи	– 1 аркуш.
– Функціональна схема системи	– 1 аркуш.
– Діаграма процесів	– 1 аркуш.
– Блок-схема алгоритму роботи програми	– 2 аркуша.
– Показники економічної ефективності	– 1 аркуш.
– Пояснювальна записка	– 91 аркуш.

10 Етапи розробки

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

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

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

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Маркетингове та економічне обґрунтування ІТ-проєкту.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

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

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

					ВКРМ-123.24.0032.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

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

_____ Усік П.С.

*Дослідження та програмна реалізація
системи KVM на основі технології HDBase-T*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 21

Літера: РП

Кропивницький – 2024 року

Основна програма

```
# Імпорт необхідних бібліотек
import socket # Для мережевої комунікації
import threading # Для використання багатопоточності
import os # Для управління системними задачами
import struct # Для роботи з бінарними даними
import time # Для додавання затримок
import cv2 # Для роботи з відеопотоками
import pickle # Для серіалізації даних
import zlib # Для компресії даних
import select # Для перевірки готовності сокетів
import sys # Системні специфічні параметри та функції

# Клас для роботи з HDBase-T технологією для KVM
class HDBaseTKVM:
    def __init__(self, ip, port):
        # Ініціалізація системи з використанням IP та порту
        self.ip = ip
        self.port = port
        self.server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

# Створення TCP сокету
self.client_socket = None # Сокет для підключення клієнта
self.clients = [] # Список підключених клієнтів
self.buffer_size = 4096 # Розмір буфера для передачі даних
self.video_stream = None # Потік відео

# Метод для запуску сервера
def start_server(self):
    # Прив'язка сокету до IP та порту
    self.server_socket.bind((self.ip, self.port))
    # Сервер слухає на вказаному порту
    self.server_socket.listen(5)
    print(f"Сервер запущено на {self.ip}:{self.port}")
    # Запуск нового потоку для обробки клієнтів
    threading.Thread(target=self.accept_clients).start()

# Метод для прийняття підключень клієнтів
def accept_clients(self):
    while True:
        # Приймаємо нове підключення
        client_socket, client_address = self.server_socket.accept()
        print(f"Клієнт {client_address} підключився")
        # Додаємо клієнта до списку
        self.clients.append(client_socket)
        # Створюємо новий потік для обробки кожного клієнта
        threading.Thread(target=self.handle_client,
args=(client_socket,)).start()

# Метод для обробки клієнта
def handle_client(self, client_socket):
    while True:
        try:
```

```

# Отримуємо дані від клієнта
data = client_socket.recv(self.buffer_size)
if data:
    # Декодуємо отримані дані
    self.process_data(client_socket, data)
else:
    # Якщо даних немає, клієнт відключився
    self.remove_client(client_socket)
    break
except Exception as e:
    # Обробка помилок
    print(f"Помилка при роботі з клієнтом: {e}")
    self.remove_client(client_socket)
    break

# Метод для обробки отриманих даних
def process_data(self, client_socket, data):
    # Декодуємо дані з байтів
    command = pickle.loads(data)
    if command['type'] == 'video':
        # Обробка відео потоку
        self.handle_video_stream(client_socket, command)
    elif command['type'] == 'keyboard':
        # Обробка клавіатурних команд
        self.handle_keyboard_input(client_socket, command)
    elif command['type'] == 'mouse':
        # Обробка миші
        self.handle_mouse_input(client_socket, command)

# Метод для видалення клієнта
def remove_client(self, client_socket):
    if client_socket in self.clients:
        self.clients.remove(client_socket)
        client_socket.close()
        print("Клієнт відключився")

# Метод для обробки відео потоку
def handle_video_stream(self, client_socket, command):
    # Передача відео потоку від сервера до клієнта
    video_frame = command['frame']
    compressed_data = zlib.compress(pickle.dumps(video_frame))
    client_socket.sendall(compressed_data)

# Метод для обробки клавіатурних команд
def handle_keyboard_input(self, client_socket, command):
    # Виведення отриманої клавіатурної команди
    print(f"Отримано клавіатурну команду: {command['key']}")

# Метод для обробки команд миші
def handle_mouse_input(self, client_socket, command):
    # Виведення отриманої команди миші
    print(f"Отримано команду миші: {command['button']}")

```

```

# Клас клієнта KVM для підключення до сервера HDBase-T
class HDBaseTKVMClient:
    def __init__(self, server_ip, server_port):
        # Ініціалізація клієнта з IP та портом сервера
        self.server_ip = server_ip
        self.server_port = server_port
        self.socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM) # TCP
сокет
        self.video_stream = None # Потік відео

    # Метод для підключення до сервера
    def connect_to_server(self):
        try:
            # Підключаємося до сервера
            self.socket.connect((self.server_ip, self.server_port))
            print(f"Підключено до сервера {self.server_ip}:{self.server_port}")
            # Запуск нового потоку для отримання даних від сервера
            threading.Thread(target=self.receive_data).start()
        except Exception as e:
            # Обробка помилок
            print(f"Помилка підключення до сервера: {e}")

    # Метод для отримання даних від сервера
    def receive_data(self):
        while True:
            try:
                # Отримуємо дані від сервера
                data = self.socket.recv(4096)
                if data:
                    # Розпаковуємо та декомпресуємо дані
                    decompressed_data = zlib.decompress(data)
                    command = pickle.loads(decompressed_data)
                    self.process_command(command)
            except Exception as e:
                # Обробка помилок
                print(f"Помилка при отриманні даних: {e}")
                break

    # Метод для обробки команд від сервера
    def process_command(self, command):
        if command['type'] == 'video':
            # Виводимо отримане відео
            self.display_video_frame(command['frame'])
        elif command['type'] == 'keyboard':
            # Обробляємо клавіатурну команду
            self.send_keyboard_input(command['key'])
        elif command['type'] == 'mouse':
            # Обробляємо команди миші
            self.send_mouse_input(command['button'])

    # Метод для відображення відео кадру
    def display_video_frame(self, frame):
        # Перетворюємо кадр в зображення та показуємо

```

```
image = pickle.loads(frame)
cv2.imshow('Video Stream', image)
cv2.waitKey(1)

# Метод для надсилання клавіатурних даних на сервер
def send_keyboard_input(self, key):
    # Створюємо команду та надсилаємо на сервер
    command = {'type': 'keyboard', 'key': key}
    self.socket.sendall(pickle.dumps(command))

# Метод для надсилання даних миші на сервер
def send_mouse_input(self, button):
    # Створюємо команду та надсилаємо на сервер
    command = {'type': 'mouse', 'button': button}
    self.socket.sendall(pickle.dumps(command))

# Головна програма
if __name__ == "__main__":
    # Створюємо сервер або клієнта в залежності від параметрів
    if len(sys.argv) > 1 and sys.argv[1] == 'server':
        server = HDBaseTKVM('0.0.0.0', 8080) # Створення сервера на порту 8080
        server.start_server() # Запуск сервера
    else:
        client = HDBaseTKVMClient('127.0.0.1', 8080) # Створення клієнта для
        підключення до локального сервера
        client.connect_to_server() # Підключення до сервера
```

Файл hdbaset_kvm_server.py

```
# Імпорт необхідних бібліотек
import socket
import threading
import pickle
import zlib
import cv2

# Клас клієнта HDBase-T для KVM
class HDBaseTKVMClient:
    def __init__(self, server_ip, server_port):
        # Ініціалізація клієнта з IP та портом сервера
        self.server_ip = server_ip
        self.server_port = server_port
        self.socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        self.buffer_size = 4096

    # Метод для підключення до сервера
    def connect_to_server(self):
        try:
            self.socket.connect((self.server_ip, self.server_port))
            print(f"Підключено до сервера {self.server_ip}:{self.server_port}")
            threading.Thread(target=self.receive_data).start()
        except Exception as e:
            print(f"Помилка підключення до сервера: {e}")

    # Метод для отримання даних від сервера
    def receive_data(self):
        while True:
            try:
                data = self.socket.recv(self.buffer_size)
                if data:
                    command = pickle.loads(zlib.decompress(data))
                    self.process_command(command)
            except Exception as e:
                print(f"Помилка при отриманні даних: {e}")
                break

    # Метод для обробки команд від сервера
    def process_command(self, command):
        if command['type'] == 'video':
            self.display_video_frame(command['frame'])
        elif command['type'] == 'keyboard':
            print(f"Клавіатурна команда: {command['key']}")
        elif command['type'] == 'mouse':
            print(f"Команда миші: {command['button']}")

    # Метод для відображення відео кадру
    def display_video_frame(self, frame):
        image = pickle.loads(frame)
        cv2.imshow('Video Stream', image)
        cv2.waitKey(1)
```

```
# Метод для надсилання клавіатурної команди на сервер
def send_keyboard_input(self, key):
    command = {'type': 'keyboard', 'key': key}
    compressed_data = zlib.compress(pickle.dumps(command))
    self.socket.sendall(compressed_data)

# Метод для надсилання команд миші на сервер
def send_mouse_input(self, button):
    command = {'type': 'mouse', 'button': button}
    compressed_data = zlib.compress(pickle.dumps(command))
    self.socket.sendall(compressed_data)

# Головна частина клієнта
if __name__ == "__main__":
    client = HDBaseTKVMClient('127.0.0.1', 8080)
    client.connect_to_server()
```

КБПЗ_2024

Файл hdbaset_kvm_client.py

```
# Імпорт необхідних бібліотек
import socket
import threading
import pickle
import zlib
import cv2

# Клас клієнта HDBase-T для KVM
class HDBaseTKVMClient:
    def __init__(self, server_ip, server_port):
        # Ініціалізація клієнта з IP та портом сервера
        self.server_ip = server_ip
        self.server_port = server_port
        self.socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        self.buffer_size = 4096

    # Метод для підключення до сервера
    def connect_to_server(self):
        try:
            self.socket.connect((self.server_ip, self.server_port))
            print(f"Підключено до сервера {self.server_ip}:{self.server_port}")
            threading.Thread(target=self.receive_data).start()
        except Exception as e:
            print(f"Помилка підключення до сервера: {e}")

    # Метод для отримання даних від сервера
    def receive_data(self):
        while True:
            try:
                data = self.socket.recv(self.buffer_size)
                if data:
                    command = pickle.loads(zlib.decompress(data))
                    self.process_command(command)
            except Exception as e:
                print(f"Помилка при отриманні даних: {e}")
                break

    # Метод для обробки команд від сервера
    def process_command(self, command):
        if command['type'] == 'video':
            self.display_video_frame(command['frame'])
        elif command['type'] == 'keyboard':
            print(f"Клавіатурна команда: {command['key']}")
        elif command['type'] == 'mouse':
            print(f"Команда миші: {command['button']}")

    # Метод для відображення відео кадру
    def display_video_frame(self, frame):
        image = pickle.loads(frame)
        cv2.imshow('Video Stream', image)
        cv2.waitKey(1)
```

```
# Метод для надсилання клавіатурної команди на сервер
def send_keyboard_input(self, key):
    command = {'type': 'keyboard', 'key': key}
    compressed_data = zlib.compress(pickle.dumps(command))
    self.socket.sendall(compressed_data)

# Метод для надсилання команд миші на сервер
def send_mouse_input(self, button):
    command = {'type': 'mouse', 'button': button}
    compressed_data = zlib.compress(pickle.dumps(command))
    self.socket.sendall(compressed_data)

# Головна частина клієнта
if __name__ == "__main__":
    client = HDBaseTKVMClient('127.0.0.1', 8080)
    client.connect_to_server()
```

КБПЗ_2024

файл hdbaset_kvmm_video.py

```
# Імпорт бібліотек
import cv2
import pickle

# Клас для захоплення та відправки відео потоку
class VideoStream:
    def __init__(self, camera_index=0):
        # Ініціалізація захоплення відео з камери
        self.capture = cv2.VideoCapture(camera_index)

    # Метод для отримання кадру з камери
    def get_frame(self):
        ret, frame = self.capture.read()
        if ret:
            # Перетворюємо кадр у байти
            return pickle.dumps(frame)
        return None

    # Метод для звільнення ресурсів камери
    def release(self):
        self.capture.release()

# Тестування відео потоку
if __name__ == "__main__":
    video_stream = VideoStream()
    while True:
        frame = video_stream.get_frame()
        if frame:
            # Відображення кадру на екрані для тестування
            cv2.imshow('Test Video Stream', pickle.loads(frame))
            if cv2.waitKey(1) & 0xFF == ord('q'):
                break
    video_stream.release()
    cv2.destroyAllWindows()
```

Файл hdbaset_kvm_keyboard.py

```
import pickle
import zlib

# Клас для обробки та передачі клавіатурних команд
class KeyboardHandler:
    def __init__(self):
        self.current_key = None

    # Метод для запису клавіатурної команди
    def record_key(self, key):
        self.current_key = key
        return self.create_command(key)

    # Створення команди для передачі на сервер
    def create_command(self, key):
        command = {'type': 'keyboard', 'key': key}
        return zlib.compress(pickle.dumps(command))

    # Метод для надсилання команди
    def send_key(self, socket, key):
        compressed_data = self.create_command(key)
        socket.sendall(compressed_data)

# Тестування функціоналу клавіатури
if __name__ == "__main__":
    keyboard_handler = KeyboardHandler()
    test_key = 'A' # Тестова клавіша
    command = keyboard_handler.record_key(test_key)
    print(f"Згенерована команда: {pickle.loads(zlib.decompress(command))}")
```

Файл hdbaset_kvm_mouse.py

```
import pickle
import zlib

# Клас для обробки та передачі подій миші
class MouseHandler:
    def __init__(self):
        self.current_button = None

    # Метод для запису подій миші
    def record_click(self, button):
        self.current_button = button
        return self.create_command(button)

    # Створення команди для передачі на сервер
    def create_command(self, button):
        command = {'type': 'mouse', 'button': button}
        return zlib.compress(pickle.dumps(command))

    # Метод для надсилання команди
    def send_click(self, socket, button):
        compressed_data = self.create_command(button)
        socket.sendall(compressed_data)

# Тестування функціоналу миші
if __name__ == "__main__":
    mouse_handler = MouseHandler()
    test_button = 'left' # Тестове натискання кнопки
    command = mouse_handler.record_click(test_button)
    print(f"Згенерована команда: {pickle.loads(zlib.decompress(command))}")
```

Файл hdbaset_kvm_error_handler.py

```
# Клас для обробки помилок
class ErrorHandler:
    @staticmethod
    def log_error(error_message):
        # Логування помилки у файл
        with open("error_log.txt", "a") as log_file:
            log_file.write(f"{error_message}\n")

    @staticmethod
    def print_error(error_message):
        # Виведення помилки на екран
        print(f"Помилка: {error_message}")

# Тестування обробки помилок
if __name__ == "__main__":
    try:
        # Тестова ситуація
        1 / 0
    except ZeroDivisionError as e:
        ErrorHandler.log_error(f"Зловлено виключення: {e}")
        ErrorHandler.print_error(f"Зловлено виключення: {e}")
```

Файл hdbaset_kvm_auth.py

```
import hashlib

# Клас для керування аутентифікацією користувачів
class UserAuth:
    def __init__(self):
        # Список користувачів та хешованих паролів
        self.users = {'admin': self.hash_password('admin123')}

    # Метод для хешування паролю
    def hash_password(self, password):
        return hashlib.sha256(password.encode()).hexdigest()

    # Метод для перевірки входу
    def authenticate(self, username, password):
        if username in self.users and self.users[username] == self.hash_password(password):
            print(f"Користувач {username} успішно аутентифікований")
            return True
        else:
            print("Невірний логін або пароль")
            return False

# Тестування аутентифікації
if __name__ == "__main__":
    auth_system = UserAuth()
    username = input("Введіть ім'я користувача: ")
    password = input("Введіть пароль: ")
    auth_system.authenticate(username, password)
```

Файл hdbaset_kvm_config.py

```
import json

# Клас для зберігання та обробки конфігурацій системи
class ConfigHandler:
    def __init__(self, config_file='config.json'):
        self.config_file = config_file
        self.config_data = self.load_config()

    # Завантаження конфігурації з файлу
    def load_config(self):
        try:
            with open(self.config_file, 'r') as f:
                return json.load(f)
        except FileNotFoundError:
            print("Файл конфігурацій не знайдено. Використовується стандартна конфігурація.")
            return {}

    # Метод для отримання параметрів
    def get(self, key, default=None):
        return self.config_data.get(key, default)

    # Метод для збереження конфігурації
    def save_config(self):
        with open(self.config_file, 'w') as f:
            json.dump(self.config_data, f, indent=4)

# Тестування роботи з конфігураціями
if __name__ == "__main__":
    config = ConfigHandler()
    print("IP сервера:", config.get('server_ip', '127.0.0.1'))
    print("Порт сервера:", config.get('server_port', 8080))
```

Файл hdbaset_kvm_logger.py

```
import logging

# Налаштування системи логування
logging.basicConfig(filename='kvm_system.log', level=logging.DEBUG,
format='%(asctime)s - %(levelname)s - %(message)s')

# Клас для логування подій в системі
class Logger:
    @staticmethod
    def log_info(message):
        logging.info(message)

    @staticmethod
    def log_warning(message):
        logging.warning(message)

    @staticmethod
    def log_error(message):
        logging.error(message)

# Тестування системи логування
if __name__ == "__main__":
    Logger.log_info("Сервер запущений")
    Logger.log_warning("Низький рівень пам'яті")
    Logger.log_error("Втрата з'єднання з клієнтом")
```

hdbaset_kvm_license.py

```

# Ліцензія та авторські права для системи HDBase-T KVM
class LicenseInfo:
    def __init__(self):
        self.license_text = """
        © 2024 Усі права захищені.
        Ліцензія: Freeware
        Ви маєте право:
        - Використовувати цю програму безкоштовно для особистих і комерційних цілей.
        - Копіювати та розповсюджувати програмне забезпечення без змін, включаючи
        збереження цієї ліцензії.
        - Використовувати програмне забезпечення в освітніх, дослідницьких та інших
        некомерційних цілях.

        Ви НЕ маєте права:
        - Продавати програму або її частини, за винятком домовленості з автором.
        - Модифікувати код і поширювати його як власну розробку без вказання
        першоджерела.
        - Вимагати будь-яку винагороду за використання програмного забезпечення, крім
        домовленості з автором.

        ВІДМОВА ВІД ГАРАНТІЙ:
        ЦЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ НАДАЄТЬСЯ "ЯК Є", БЕЗ БУДЬ-ЯКИХ ЯВНИХ ЧИ НЕЯВНИХ
        ГАРАНТІЙ, ВКЛЮЧАЮЧИ, АЛЕ НЕ ОБМЕЖУЮЧИСЬ, ГАРАНТІЯМИ ПРИДАТНОСТІ ДЛЯ КОНКРЕТНИХ
        ЦІЛЕЙ. АВТОР НЕ НЕСЕ ВІДПОВІДАЛЬНОСТІ ЗА БУДЬ-ЯКІ ШКОДИ, ЩО ВИНΙΚАЮТЬ В
        РЕЗУЛЬТАТІ ВИКОРИСТАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ, НАВІТЬ ЯКЩО АВТОР БУВ
        ПОПЕРЕДЖЕНИЙ ПРО МОЖЛИВІСТЬ ТАКИХ ШКОД.
        """

        # Метод для відображення ліцензійної інформації
        def display_license(self):
            print(self.license_text)

# Тестування ліцензії
if __name__ == "__main__":
    license_info = LicenseInfo()
    license_info.display_license()

```

```
import psutil

class SystemMonitor:
    @staticmethod
    def get_cpu_usage():
        return psutil.cpu_percent(interval=1)

    @staticmethod
    def get_memory_usage():
        memory = psutil.virtual_memory()
        return memory.percent

    @staticmethod
    def get_disk_usage():
        disk = psutil.disk_usage('/')
        return disk.percent

# Тестування моніторингу
if __name__ == "__main__":
    print(f"Використання процесора: {SystemMonitor.get_cpu_usage()}%")
    print(f"Використання пам'яті: {SystemMonitor.get_memory_usage()}%")
    print(f"Використання диска: {SystemMonitor.get_disk_usage()}%")
```

Файл hdbaset_kvm_web.py

```
from flask import Flask, render_template
from hdbaset_kvm_config import ConfigHandler
from hdbaset_kvm_monitor import SystemMonitor

app = Flask(__name__)
config = ConfigHandler()

@app.route('/')
def index():
    server_ip = config.get('server_ip')
    server_port = config.get('server_port')
    cpu_usage = SystemMonitor.get_cpu_usage()
    memory_usage = SystemMonitor.get_memory_usage()
    disk_usage = SystemMonitor.get_disk_usage()
    return render_template('index.html',
                           server_ip=server_ip,
                           server_port=server_port,
                           cpu_usage=cpu_usage,
                           memory_usage=memory_usage,
                           disk_usage=disk_usage)

if __name__ == "__main__":
    app.run(host='0.0.0.0', port=5000)

<!DOCTYPE html>
<html>
<head>
    <title>KVM System Monitor</title>
</head>
<body>
    <h1>Стан системи KVM</h1>
    <p>IP сервера: {{ server_ip }}</p>
    <p>Порт сервера: {{ server_port }}</p>
    <h2>Моніторинг системи:</h2>
    <p>Використання CPU: {{ cpu_usage }}%</p>
    <p>Використання пам'яті: {{ memory_usage }}%</p>
    <p>Використання диска: {{ disk_usage }}%</p>
</body>
</html>
```

Файл hdbaset_kvm_backup.py

```
import shutil
import os
import datetime

class BackupManager:
    def __init__(self, config_file='config.json', backup_dir='backups'):
        self.config_file = config_file
        self.backup_dir = backup_dir
        if not os.path.exists(self.backup_dir):
            os.makedirs(self.backup_dir)

    # Метод для створення резервної копії
    def create_backup(self):
        timestamp = datetime.datetime.now().strftime('%Y%m%d_%H%M%S')
        backup_file = os.path.join(self.backup_dir,
f"config_backup_{timestamp}.json")
        shutil.copy(self.config_file, backup_file)
        print(f"Резервну копію збережено як {backup_file}")

# Тестування функціоналу резервного копіювання
if __name__ == "__main__":
    backup_manager = BackupManager()
    backup_manager.create_backup()
```

Файл hdbaset_kvmm_encrypt.py

```
from cryptography.fernet import Fernet

class ConfigEncryptor:
    def __init__(self, key_file='secret.key'):
        self.key_file = key_file
        self.key = self.load_key()

    # Метод для генерації та збереження ключа

    def generate_key(self):
        key = Fernet.generate_key()
        with open(self.key_file, 'wb') as key_file:
            key_file.write(key)

    # Завантаження ключа
    def load_key(self):
        with open(self.key_file, 'rb') as key_file:
            return key_file.read()

    # Шифрування файлу конфігурації
    def encrypt_file(self, file_name):
        with open(file_name, 'rb') as file:
            data = file.read()
        f = Fernet(self.key)
        encrypted_data = f.encrypt(data)
        with open(file_name + '.enc', 'wb') as encrypted_file:
            encrypted_file.write(encrypted_data)
        print(f"Файл {file_name} зашифровано.")

    # Розшифрування файлу конфігурації

    def decrypt_file(self, encrypted_file_name):
        with open(encrypted_file_name, 'rb') as encrypted_file:
            encrypted_data = encrypted_file.read()

        f = Fernet(self.key)
        decrypted_data = f.decrypt(encrypted_data)

        with open(encrypted_file_name.replace('.enc', ''), 'wb') as
decrypted_file:
            decrypted_file.write(decrypted_data)
        print(f"Файл {encrypted_file_name} розшифровано.")

    # Тестування шифрування
    if __name__ == "__main__":
        encryptor = ConfigEncryptor()
        encryptor.generate_key()
        encryptor.encrypt_file('config.json')
        encryptor.decrypt_file('config.json.enc')
```