

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

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
« ____ » _____ 2026 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за першим (бакалаврським) рівнем вищої освіти
на тему
“Програмне забезпечення системи Legrand Cabling System 3 для ЦОД”

Виконав здобувач вищої освіти
IV курсу, групи KI-22-1
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Пономаренко С.С.
« ____ » _____ 2026 р.

Керівник проекту
доктор філософії (PhD)
_____ Дреєва Г.М.
« ____ » _____ 2026 р.

Рецензент _____

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

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 27 » лютого 2026 року

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

Пономаренку Сергію Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Програмне забезпечення системи Legrand Cabling System 3 для ЦОД

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

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

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

3. Строк подання студентом роботи до захисту 23.05.2026 р.

4. Мета та завдання випускної кваліфікаційної роботи: Метою роботи є розробка програмного забезпечення системи Legrand Cabling System 3 для ЦОД

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

1. Призначення та область використання.

2. Перегляд аналогічних існуючих систем.

3. Опис і обґрунтування проектних рішень.

4. Етапи програмування системи.

5. Впровадження системи в промислову експлуатацію.

6. Висновки

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

Структурна схема системи 1 аркуш

Функціональна схема системи 1 аркуш

Діаграма процесів 1 аркуш

Блок-схема алгоритму роботи додатку 2 аркуша

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

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.03.2026 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2026 р.	
3.	Розробка моделі компонента	20.03.2026 р.	
4.	Розробка структур даних	25.03.2026 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.03.2026 р.	
6.	Програмування алгоритмів	10.04.2026 р.	
7.	Оформлення ПЗ	17.04.2026 р.	
8.	Попередній захист роботи	23.05.2026 р.	

Дата видачі завдання
« 27 » лютого 2026 р.

Підпис керівника

Дреєва Г.М.
(прізвище та ініціали)

Завдання прийнято до виконання
« 27 » лютого 2026 р.

Підпис здобувача

Пономаренко С.С.
(прізвище та ініціали)

АНОТАЦІЯ

Пономаренко С.С. Програмне забезпечення системи Legrand Cabling System 3 для ЦОД. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2026.

В даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи Legrand Cabling System 3 для ЦОД.

Метою розробки є програмне забезпечення системи Legrand Cabling System 3 для ЦОД.

Результат роботи – програмна реалізація системи Legrand Cabling System 3 для ЦОД.

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, Legrand Cabling System 3, ЦОД

ABSTRACT

Ponomarenko S.S. Software for the Legrand Cabling System 3 system for data centers. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2026.

In this final qualification work for the first (bachelor's) level of higher education, software has been developed that is intended for the Legrand Cabling System 3 system for data centers.

The purpose of the development is the software for the Legrand Cabling System 3 system for data centers.

The result of the work is the software implementation of the Legrand Cabling System 3 system for data centers.

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 are provided.

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

The program was developed in the Python environment.

Keywords: computer engineering, Legrand Cabling System 3, data center

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	2
ВСТУП.....	3
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	4
1.1 Призначення системи.....	4
1.2 Область застосування.....	5
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	6
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.....	6
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	10
2.3 Розгорнута постановка завдання	15
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	17
3.1 Опис функціонування системи	17
3.2 Розробка структурної схеми.....	21
3.3 Розробка функціональної схеми	29
3.4 Розробка діаграми процесів.....	32
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	34
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	34
4.2 Захист розробленого програмного забезпечення.....	50
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	54
6 ОСНОВНІ ВИСНОВКИ.....	60
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	62

					ВКРБ-123.26.0016.00.00.ПЗ			
Вим.	Арк.	№ докум.	Підп.	Дата	<i>Програмне забезпечення системи Legrand Cabling System 3 для ЦОД</i>	Лім.	Аркуш	Аркушів
<i>Розроб.</i>		Пономаренко С.С.				Б	1	69
<i>Перев.</i>		Дресва Г.М.						
<i>Н.контр.</i>		Коваленко А.С.				ЦНТУ КІ-22-І		
<i>Затв.</i>		Смірнов О.А.						

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

БД	—	база даних
ЛОМ	—	локальна обчислювальна мережа
РП	—	розподільчий пункт
СКС	—	структурована кабельна система
ASP	—	Active Server Pages – активні серверні сторінки
DHCP	—	Dynamic Host Configuration Protocol – протокол динамічної конфігурації вузла
HTTP	—	HyperText Transfer Protocol – протокол передачі гіпер тексту
IMAP	—	Internet Message Access Protocol – протокол доступу до електронної пошти Інтернету
ICMP	—	Internet Control Message Protocol – міжмережний протокол керуючих повідомлень
MMC	—	Microsoft Management Console
POP3	—	Post Office Protocol Version 3 – протокол поштового відделення, версія 3
SQL	—	Structured Query Language – мова структурованих запитів
SMTP	—	Simple Mail Transfer Protocol – простий протокол передачі пошти
SNMP	—	Simple Network Management Protocol – простий протокол керування мережею
Syslog	—	стандарт відправки повідомлень про зміни які відбуваються в мережі
UDP	—	User Datagram Protocol – протокол користувальницьких дейтаграм

ВСТУП

Актуальність теми. Завдяки великим даним, хмарним обчисленням та Інтернету речей, інфраструктури стали нервовим центром бізнесу та його економічного зростання. За підтримки таких партнерів, як Minkels, SJ Manufacturing та Ortronics, Legrand здатна вирішити всі ключові проблеми центрів обробки даних, тобто енергоефективність, безперервність обслуговування, безпека, гнучкість та адаптивність до змін у технологіях, правилах або використанні. Група надає технічну підтримку на кожному етапі проекту центру обробки даних, від проектування до встановлення та експлуатаційного обслуговування.

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи Legrand Cabling System 3 для ЦОД.

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

- Огляд існуючих систем Legrand Cabling System 3 для ЦОД.
- Дослідження системи Legrand Cabling System 3 для ЦОД.
- Програмна реалізація системи Legrand Cabling System 3 для ЦОД.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі Legrand Cabling System 3 для ЦОД.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи Legrand Cabling System 3 для ЦОД, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					ВКРБ-123.26.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Ще десять років тому на мідні кабельні системи доводилося близько 90% всіх з'єднань у ЦОД. Однак потреба в підтримки все зростаючих швидкостей і зниження ціна оптичні рішення (насамперед активне встаткування) привели до того, що оптика усе ширше застосовується в центрах обробки даних. Волоконо-оптична система включає повний комплекс рішень – від коннекторів і кабелів до інструментів. Один тільки асортименти оптичних кабелів включає 113 позицій.

З'єднувачі представляють основу будь-якої кабельної системи, оскільки саме від них у першу чергу залежать її характеристики В LCS3 з'явилося відразу два нових удосконалений коннектора – оптичний MTP і мідний RJ-45. Як затверджують в Legrand, MTP є самим швидкісним на ринку, а рівень внесених перешкод в оптичних касетах на його основі не перевищує 0,35 дБ. Legrand запатентувала конструкцію нового RJ-45, що може встановлюватися на кабель без інструмента. Він розрахований на застосування в системах Категорій 5e, 6, 6A и 8 і передачу живлення PoE++.

В LCS3 з'явилися нарешті-те полки високої щільності, причому як оптичні, так і мідні. Legrand пропонує оптичні полки високої й надвисокої щільності з коннекторами LC на 96 портів і 114 портів і мідні полки високої щільності на 48 порто висотою 1U. Крім того, компанія пропонує модульні касетні полки, у які можна встановлювати як оптичні, так і мідні касети. Можливість безінструментальної фіксації й добування полиці із шафи /стійки дозволяє прискорити процес обслуговування й відмовитися від використання монтажних гвинтів.

					ВКРБ-123.26.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

1.2 Область застосування

Серверні й мережні шафи мають єдиний конструктив і мають високу навантажувальну здатність (до 1,5 т), при цьому вони важать на 20% легше, ніж аналоги. Крім шаф в асортименти також входять стійки, настінні шафи й універсальні аксесуари до них. Для ізоляції потоків холодного й гарячого повітря Legrand пропонує власну запатентовану модульну систему. Як затверджується, організація холодного коридору Next Generation Corridor дозволяє заощаджувати до 30% енергії на охолодженні.

За останні кілька років Legrand придбала декілька виробників і постачальників блоків розподілу живлення – Raritan, Server Technology і Shenzhen Clever Electronic – і тепер може запропонувати різноманітний асортименти інтелектуальних PDU – горизонтальні й вертикальні блоки, трифазні, однофазні, на 16 і 32 А. Монтаж PDU може вироблятися без інструментів.

Лінійка включає три моделі. Моделі з посиленням захистом мають автоматичний вимикач для захисту встаткування від перевантажень і коротких замикань, моделі з обмежником перенапруги забезпечують захисту від імпульсних стрибків, а модель із амперметром відображає доступну потужність у шафі. Система захисту від випадкового від'єднання шнурів CoreLocking System дозволяє запобігти ненавмисному відключенню навантаження.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи Legrand Cabling System 3 для ЦОД, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					ВКРБ-123.26.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

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

У світі існує величезна кількість виробників устаткування для СКС. Це такі відомі німецькі й американські компанії як Siemon, Panduit, Shneider, GE на які так люблять будувати мережі. Це й величезна кількість виробників з Китаю й України, що часто маскуються під солідний бренд – а на ділі, що випускають алюмінієвий (замість мідного) кабель UTP непостійної товщини.

Hyperline

Асортименти продукції являє собою повний набір пасивних елементів, необхідних для побудови СКС: комутаційні панелі, розетки, волоконо-оптичні сполучні шнури, кроссове встаткування, модулі Keystone Jack, патч-корди, роз'єми – це лише невеликий перелік компонентів, що поставляються.

Розмаїтість продукції Hyperline забезпечує користувачам можливість прокладати СКС будь-якої складності. Висока якість і широкий спектр компонентів гарантує клієнтам стабільну роботу системи, а також повну сумісність зі СКС інших виробників. При цьому на всю продукцію Hyperline, включаючи компоненти категорії 6, підтримуються конкурентоспроможні ціни.

Інша настільки ж важлива частина продуктової лінійки Hyperline – це виробу із пластмаси, необхідні для маркування, структурування, захисту кабельних трас і компонентів СКС у цілому. Широкий спектр продукції із пластмаси представлений різними видами стяжок, хомутів, пластикових скоб, що самоклеяться площадок, кабельних вводів і т.п. При цьому асортименти даних виробів незмінно поповнюється новими моделями. Здійснюється повний контроль якості на всіх стадіях розробки й виробництва продукції, починаючи з

					ВКРБ-123.26.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

підготовки сировини для забезпечення відповідності кожного компонента внутрішнім і міжнародним стандартам і закінчуючи готовими виробами. Компанія Hyperline є членом міжнародних організацій і має міжнародні сертифікати на продукцію.

ZPAS

Продукція компанії ZPAS знаходить застосування в таких галузях, як телекомунікації, енергетика, гірничодобувна промисловість, металургія, хімічна промисловість, охорона навколишнього середовища, а також в областях, що вимагає використання сучасних методів автоматизації й модернізації виробничих процесів.

Монтажні шафи й стійки виробництва ZPAS використовуються на безлічі підприємств у Польщі.

Продукція компанії ZPAS користується попитом і за межами Польщі. Так, устаткуванням виробництва ZPAS оснащені електростанції в Туреччині, Саудівській Аравії, Китаї. Компанія також поставляє за договором у Німеччину серію компонентів для розподільних шаф, комутаційні панелі для залізниці, а також пульти керування й корпуси, виготовлені з нержавіючої сталі.

Legrand

Компанія Legrand – світовий лідер у виробництві продукції електротехнічного призначення. Компанія поєднує понад 22000 співробітників на п'яти континентах, має 31 філію й більше 20 централізованих офісів в усьому світі.

Legrand робить широкий спектр устаткування для офісів, промислових і житлових приміщень: пластикові й алюмінієві кабель-канали, настановні аксесуари до них, електроустановочні вироби, монтажні аксесуари, силове встаткування, компоненти систем автоматичного керування електроживленням.

Продукція Legrand відповідає найвищим вимогам і стандартам європейської якості. Основні характерні риси встаткування Legrand – бездоганна надійність, зручність і простота монтажу, безпека, довговічність, сучасний

					ВКРБ-123.26.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

дизайн. Виробляється з використанням новітніх технологій і досягнень в області електротехніки. На українському ринку Legrand працює більше 35 років. Із самого початку роботи компанії на радянському, (потім і українському ринку) виробу, вироблені Legrand, були рекомендовані до установки тільки на елітних об'єктах і об'єктах, що мають стратегічне значення, тому що завжди випереджали час за всіма показниками, пропонованим до виробів низьких напруг. Сьогодні виробу Legrand установлюють як в урядових закладах, так і житлових приміщеннях, на промислових підприємствах і офісах.

Siemon

The Siemon Company, заснована в 1903, початку свою діяльність із розробки состава матеріалу й технології виробництва ручок для столових ножів. Незабаром з'ясувалося, що цей матеріал може витримувати вплив високих температур, води що кипить, й має гарні діелектричні властивості. Це дало можливість компанії Siemon укласти контракт із AT&T (Western Electric) на виробництво 3-полюсних комутаційних блоків, які й стали її першим компонентом, зробленим для телекомунікаційних кабельних систем. Незважаючи на це, компанія кілька десятиліть випускала продукцію, що не має відносини до кабельних систем, – грамофонні пластинки й столові прилади, при цьому підтримуючи контракт із AT&T на виробництво комутаційних блоків.

Пізніше, в 60-х The Siemon Company придбала в AT&T ліцензію на технологію виробництва комутаційного встаткування на основі блоків типу 66, що й привело компанію на ринок кабельних компонентів. Компанія вирішила сконцентрувати свою діяльність на виробництві пасивних телекомунікаційних компонентів, зробивши кабельні системи й коннектори своїм основним бізнесом. З тих пор The Siemon Company випустила кілька ліній комутаційного встаткування, компонентів кабельних систем, тестуючих приладів і створила інженерно-технічний підрозділ, у завдання якого входить розробка нової передової продукції й сприяння в проектуванні й монтажі кабельних систем Siemon Cabling System.

					ВКРБ-123.26.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

Сьогодні The Siemon Company проектує й робить найбільш широкий діапазон продукції для комутації, захисту, адміністрування, маркування й тестування кабельних систем на основі UTP і ScTP 100 Ом і оптичного волокна. Компанія володіє більш ніж 200 діючими патентами у своїй області, підтверджуючи цим прагнення до світового технологічного лідерства. У число основних ліній продукції компанії Siemon входять три типи комутаційних блоків – S110, S66 і Multiflex, а також безліч ліній модульної сполучної продукції, включаючи серії CT, HD5, SM і нову серію MAX.

The Siemon Company здійснює контроль якості на всіх стадіях розробки й виробництва продукції, починаючи з підготовки сировини для забезпечення відповідності або переваги кожним компонентом внутрішніх і міжнародних стандартів. Кабельна система Siemon Cabling System підтримує діючі й нові мережні додатки, включаючи передачу мови, даних, відеозображень, TP-PMD, 100 BASE-T, Gigabit Ethernet, ATM 155 Мбіт/с і ATM 622 Мбіт/с.

Розмаїтість продукції марки Siemon і її "відкрита" конструкція дає користувачам можливість вибирати кабелі й компоненти, ґрунтуючись на їхній ціні й робочих характеристиках, на відміну від інших систем, де потрібне проходження твердій схемі й пов'язаної з нею лінії компонентів. Компанія поставляє на ринок кілька ліній продукції, що відповідає вимогам стандартів ANSI / TIA / EIA-568-A і ANSI / TIA / EIA 606, ISO / IEC 11801 і інших промислових стандартів.

Molex

Molex Incorporated – світовий лідер по виробництву електронних, електричних і волоконо-оптичних компонентів, а також комплексних рішень в області СКК (підрозділ Molex Premise Networks). У числі партнерів компанії такі фірми як CISCO, Ford, Intel, Nokia, Phillips, Siemens, 3Com і інші. Компанія Molex заснована в 1938 році в США (штат Іллінойс), на сьогоднішній день має оберт в 3 млрд. доларів США. 55 заводів і фабрик компанії розташовані в 19 країнах миру, на них працює більше 32000 чоловік.

					ВКРБ-123.26.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

У липні 1994 року корпорація Molex придбала компанію Mod-Tap з метою більше активної участі в розвитку й просуванні на ринок рішень для структурованих кабельних систем. Це злиття дало можливість компанії Molex розширити спектр своїх продуктів і пропонувати не тільки окремі компоненти виробництва Molex, але й комплексні рішення від Molex Premise Networks.

Корпорація Molex приділяє особливу увагу ринкам Східної Європи, де одним з перспективних є ринок України.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Python – динамічна інтерпретована об’єктно-орієнтована скриптова мова програмування із строгою динамічною типізацією. Офіційний сайт мови програмування Python <https://www.python.org/>. Python – багатоцільова мова програмування, яка дозволяє писати код, що добре читається. Відносний лаконізм мови Python дозволяє створити програму, яка буде набагато коротше свого аналога, написаного на іншій мові. Python – багатоплатформова мова програмування. Це означає, що програми на Python можна запускати в різних операційних системах без будь-яких змін.

Ще однією перевагою Python є його стандартна бібліотека, яка встановлюється разом з Python і містить готові інструменти для роботи з операційною системою, веб-сторінками, базами даних, різними форматами даних, для побудови графічного інтерфейсу програм тощо. Програми, написані на мові програмування Python, можуть бути як невеликими скриптами, так і складними системами. Python абсолютно безкоштовний.

Швидкість виконання коду Python

Один з можливих недоліків Python – швидкість виконання коду. Python не є компільованою мовою. Код на Python спочатку компілюється у внутрішній байт-код, який потім виконується інтерпретатором Python. У більшості випадків

					ВКРБ-123.26.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

при використанні Python виходять програми повільніші в порівнянні з такими мовами, як С.

Втім, сучасні комп'ютери мають таку обчислювальну потужність, що для більшості застосунків швидкість розробки важливіша швидкості виконання, а програми на Python зазвичай пишуться набагато швидше.

Окрім того, Python легко розширюється модулями, написаними на С або С++. Такі модулі можуть використовуватися для виконання частин програми, що створюють інтенсивне навантаження на процесор.

Використання Python

Python використовується для різних цілей: для створення ігор і веб-застосунків, розробки внутрішніх інструментів для різноманітних проектів. Мова також широко застосовується в науковій області для досліджень і розв'язування прикладних завдань.

Застосування мови програмування Python:

1. BitTorrent – протокол для обміну даними.
2. Ubuntu Software Center – вільне програмне забезпечення для пошуку, установки і видалення пакунків в системі Ubuntu Linux.
3. Blender – програма для створення тривимірної комп'ютерної графіки, що включає засоби моделювання, анімації, вимальовування, пост-обробки відео, а також створення відеоігор.
4. GIMP – растровий графічний редактор, із підтримкою векторної графіки.
5. World of Tanks.
6. Вільна енциклопедія Вікіпедія.
7. Пошукова система Google.
8. DropBox – файловий хостинг, що включає персональне хмарне сховище, синхронізацію файлів і програму-клієнт.
9. YouTube – популярне відеосховище.

Версії Python

					ВКРБ-123.26.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

Мови програмування з часом змінюються – розробники додають в них нові можливості, а також виправляють помилки. Так з’являються різні версії мови. Наприклад, код написаний на Python 2 у більшості випадків не буде працювати у версії Python 3 без внесення додаткових змін.

Процесор є найважливішим компонентом в комп’ютері. Одна з основних функцій процесора – це обробка даних згідно комп’ютерної програми, яка є списком інструкцій, шляхом виконання арифметичних і логічних операцій над фрагментами даних.

Кожна інструкція в програмі – це команда, яка «повідомляє» процесору, яку операцію він повинен виконати. Процесор комп’ютера може розуміти лише ті інструкції, які написані на машинній мові. Машинна мова – це штучна мова, створена для передачі команд комп’ютеру. За допомогою машинної мови створюються ефективні програми, оскільки розробник отримує доступ до всіх можливостей процесора. Машинна мова – мова низького рівня.

Інструкція машинної мови існує для кожної операції, яку процесор здатний виконати – є інструкція для додавання чисел, є інструкція для віднімання чисел і т.д. Увесь набір інструкцій, який центральний процесор може виконати, відомий як набір інструкцій процесора.

Наприклад, у вас є певна програма, яка зберігається на диску вашого комп’ютера. Для виконання програми, ви здійснюєте подвійний клік на значку програми. Це змушує програму копіюватися з диска в оперативну пам’ять, після чого процесор комп’ютера виконує копію програми, яка знаходиться в оперативній пам’яті.

Коли процесор виконує інструкції програми, він бере участь у процесі, який є відомим як цикл `fetch – decode – execute` (отримати – декодувати – виконати). Цей цикл виконується для кожної інструкції у програмі і складається з трьох кроків:

Отримати

Програма – це послідовність інструкцій на машинній мові. Першим кроком циклу є завантаження (отримання) наступної інструкції з пам'яті в процесор.

Декодувати

Інструкція машинної мови – це двійкове число, яке представляє команду, що повідомляє процесору виконати певну операцію. На цьому кроці процесор декодує інструкцію, яку було «витягнуто» з пам'яті, для визначення того, яка операція повинна виконуватись.

Виконати

Останній крок циклу – виконати операцію.

Хоча процесор комп'ютера розуміє тільки машинну мову, людині непрактично писати програми на машинній мові. Така програма може мати тисячі або навіть мільйони бінарних інструкцій, і написання такої програми буде дуже обтяжливим процесом.

З цієї причини була створена мова асемблера як альтернатива машинній мові. Замість використання двійкових чисел для написання інструкцій, мова асемблера використовує короткі слова, відомі як мнемокоди.

Незважаючи на те, що мова асемблера не вимагає двійкових інструкцій, як у випадку машинної мови, проте вона вимагає високих знань про процесор. Використовуючи мову асемблера, навіть для найпростішої програми, необхідно написати велику кількість інструкцій.

Мова програмування високого рівня дозволяє створювати складні програми, не знаючи, як працює процесор, і не записуючи великої кількості інструкцій низького рівня. Крім того, більшість мов програмування високого рівня використовують слова, які легко зрозуміти.

Python – одна із популярних сучасних мов програмування високого рівня. Python – інтерпретована мова програмування. Python – це високорівнева інтерпретована мова програмування, на відміну від C++, яка є прикладом

					ВКРБ-123.26.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

компільованої мови програмування. Назва Python відноситься як до мови програмування, так і до інтерпретатора – комп'ютерної програми, яка зчитує початковий код (написаний на Python) і виконує інструкції (команди).

Для перекладу мови високого рівня на машинну мову доступні два типи програм:

1. Компілятор.
2. Інтерпретатор.

Завантаження Python

Версії інтерпретатора Python для різних операційних систем доступні для безкоштовного завантаження за адресою <https://www.python.org/downloads>.

Середовище програмування для Python

Для написання програм використовують текстові редактори або інтегровані середовища розробки, які включають в себе різні інструменти для роботи з кодом: засіб для написання коду (текстовий редактор), інтерактивний інтерпретатор, відлагоджувач тощо.

Текстові редактори та інтегровані середовища програмування для Python:

- IDLE – стандартний редактор Python. Встановлюється разом з Python для користувачів Windows, окремим пакунком для користувачів Linux.
- Notepad++ – безкоштовний текстовий редактор початкового коду, який підтримує велику кількість мов, в тому числі і Python. Лише для користувачів Windows.
- Visual Studio Code – це легкий, але потужний редактор початкового коду, який розповсюджується безкоштовно і доступний у версіях для платформ Linux, Windows і macOS.
- PyScripter – інтегроване середовище розробки для мови програмування Python. Для користувачів Windows. Поширюється безкоштовно.
- Wing IDE 101 – вільне інтегроване середовище для Python, розроблене для навчання програмістів-початківців. Для користувачів Linux, Windows і macOS. Поширюється безкоштовно.

					ВКРБ-123.26.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

– Geany – вільний текстовий редактор з базовими елементами інтегрованого середовища розробки, доступний для операційних систем Linux, Windows і macOS.

– PyCharm – інтегроване середовище розробки для мови програмування Python. PyCharm є власницьким програмним забезпеченням. Наявна безкоштовна версія Community з усіченим набором можливостей. Для користувачів Linux, Windows і macOS.

– Thonny – IDE для вивчення програмування мовою Python. Для користувачів Linux, Windows і macOS.

– Mu – редактор коду Python для програмістів-початківців. Для користувачів Linux, Windows і macOS.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи Legrand Cabling System 3 для ЦОД.

В процесі розробки випускної кваліфікаційної роботи за першим (бакалаврським) 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

					ВКРБ-123.26.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

в роботі технологічного обладнання;

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

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

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

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

КБПЗ-2026

					ВКРБ-123.26.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Модульний ДБЖ Legrand Keor FLEX

На відміну від традиційних монолітних систем ДБЖ, які потребують повної заміни системи після закінчення терміну служби, Keor FLEX був розроблений з самого початку для продовження життєвого циклу та модульного повторного використання.

Його справжня модульна архітектура з можливістю гарячої заміни дозволяє незалежно замінювати, відновлювати або перерозподіляти окремі модулі живлення або байпасу, що значно зменшує кількість відходів після закінчення терміну служби та подовжує загальний термін служби системи.

Дизайн системи «оплата за зростання» дозволяє операторам розгортати лише ті потужності, які їм потрібні, саме тоді, коли вони їм потрібні. Це дозволяє уникнути надмірного виділення ресурсів, зменшує викиди вуглецю та запобігає блокуванню інфраструктурних активів.

Keor FLEX досягає:

- ККД 98,6% в режимі онлайн-подвійного перетворення (сертифіковано TÜV).
- ККД понад 99% в еко-режимі.
- Коефіцієнт переробки 85% (IEC/TR 62635).
- Понад 69% металу, придатного для вторинної переробки.
- 50% упаковки, що містить перероблену продукцію.
- 1,2 МВт на площі всього 2,8 м².

Завдяки інтеграції передової технології карбіду кремнію (SiC) та низькоомної архітектури внутрішніх шин, Keor FLEX зменшує теплове навантаження та подовжує термін служби напівпровідникових і силових модулів,

					ВКРБ-123.26.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

забезпечуючи довше та надійніше повторне використання.

Розроблено для зменшення відходів у всій системі

Keor FLEX розроблений для зменшення відходів протягом життєвого циклу, уникаючи непотрібної заміни акумуляторів. Його універсальний інтерфейс для акумуляторів дозволяє операторам зберігати та повторно використовувати існуючі VRLA, літій-іонні або нікель-цинкові акумуляторні блоки під час модернізації, зменшуючи передчасну утилізацію та негативний вплив нових акумуляторних блоків.

Його внутрішня архітектура шин зменшує кількість кабельних з'єднань та поширених точок теплового зриву, а розподілена конструкція стійкості, перевірена за допомогою моделювання FMECA, обмежує непотрібну заміну модулів та продовжує термін служби.

Система також підтримує функціональність інтелектуальної мережі, включаючи зменшення пікових навантажень та швидку частотну характеристику, що допомагає центрам обробки даних знизити навантаження на мережу та захистити інфраструктуру вищезгаданих мереж від стрибків потужності, спричинених штучним інтелектом.

Крім того, Keor FLEX пройшов перевірку як готовий до використання зі штучним інтелектом, його продуктивність відповідає робочим навантаженням графічних процесорів високої щільності, а також було протестовано в умовах навантаження на штучний інтелект відповідно до стандартів IEC 62040-3.

Сталий розвиток без шкоди для продуктивності

Центри обробки даних перебувають під тиском, щоб забезпечити більшу обчислювальну потужність, одночасно зменшуючи вплив на навколишнє середовище.

Keor FLEX доводить, що сталий розвиток та продуктивність не є взаємовиключними. Впроваджуючи принципи циркулярності в основну архітектуру, ми створили платформу ДБЖ, яка продовжує термін служби активів, зменшує кількість відходів, знижує споживання енергії та підтримує робочі

					ВКРБ-123.26.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

навантаження зі штучним інтелектом, що розвиваються, – і все це одночасно покращує загальну вартість володіння. Це відображає ширші зобов'язання Legrand щодо сталого розвитку, зокрема отримання рейтингу «А» від CDP другий рік поспіль.

Завдяки масштабованості від 100 кВт до 1,2 МВт на корпус та до 4,8 МВт паралельно, Keog FLEX підтримує роботу від середніх об'єктів до гіпермасштабних кампусів.

Legrand: Розширення можливостей інфраструктури центрів обробки даних зі штучним інтелектом

Компанія Legrand оголосила про придбання Kratos Industries та стратегічні інвестиції в Accelsius. Ці кроки спрямовані на зміцнення її портфоліо комплексних рішень для управління живленням та температурою для центрів обробки даних, готових до використання штучного інтелекту.

Подвійне оголошення розширює можливості Legrand як у «сірій» сфері, що охоплює розподіл живлення вище за течією, так і в «білій» сфері. Саме тут розгортаються стійки високої щільності та вдосконалені системи охолодження.

Оскільки оператори масштабують інфраструктуру штучного інтелекту, інтеграція між цими рівнями стає центральною для надійності та продуктивності.

Розширення критично важливих енергосистем

Компанія Kratos Industries, що базується в Арваді, штат Колорадо, виробляє низьковольтні розподільні щити, розподільчі пристрої середньої напруги та інтегровані системи живлення для центрів обробки даних та промислового застосування.

Компанія відома своєю вертикально інтегрованою моделлю виробництва та підходом до розробки проектів на замовлення. Це забезпечує швидке постачання для критично важливих середовищ.

Придбання Kratos розширює існуючий портфель Legrand для виробництва "сірих просторів", включаючи кабельні шини, навантажувальні блоки та системи кабельних лотків. Це доповнює пропозиції Legrand для виробництва "білих

просторів", такі як шинопроводи, шинопроводи та стійкові розподільні пристрої живлення.

Проекти центрів обробки даних дедалі більше потребують інтегрованих партнерів для підтримки всього ланцюга живлення. Це охоплює від взаємоз'єднання комунальних служб до розподілу на рівні стійок.

Ця інтеграція є життєво важливою, оскільки постачальники послуг гіпермасштабування та колокації стикаються з безпрецедентним попитом.

Центри обробки даних стикаються з безпрецедентним тиском щодо побудови цифрової інфраструктури у великих масштабах та з надзвичайною надійністю.

Консолідація постачальників може спростити закупівлі та зменшити ризик інтеграції у масштабних будівлях. Це особливо актуально, оскільки проекти рухаються до гігаватних кампусів, що потребують надійних промислових енергетичних рішень.

Підтримка складних інфраструктурних проектів

Використовуючи глобальний ланцюг поставок та операційну інфраструктуру Legrand, ми можемо пришвидшити зростання та краще підтримувати масштабні, складні проекти, які реалізують наші клієнти.

Для розвитку інфраструктури, пошук узгоджених енергетичних рішень від одного постачальника пропонує переваги у стандартизації проектування та швидкості розгортання.

Удосконалені рішення для термоменеджменту

Legrand також взяла участь у раунді фінансування серії В для Accelsius, спеціаліста з двофазного рідинного охолодження безпосередньо до мікросхем. Це партнерство розширює присутність Legrand у сфері передового теплового управління зі збільшенням щільності стійок.

Робочі навантаження, пов'язані зі штучним інтелектом та високопродуктивними обчисленнями, виштовхують традиційне повітряне охолодження за межі практичних можливостей. Це прискорює перехід до

рідинних систем.

Технологія двофазного охолодження Accelsius відводить тепло безпосередньо на рівні процесора. Використовуючи принципи фазової зміни, система забезпечує вищу щільність обчислень, одночасно зменшуючи потреби в енергії для охолодження.

Legrand та Accelsius співпрацюватимуть над спільними ініціативами з розвитку у сфері вільного простору. Це включає інтеграцію рідинного охолодження в стійкову інфраструктуру.

Ця співпраця відображає ширшу галузеву тенденцію, в якій збігаються аспекти живлення, механіки та проектування стійок. Інтеграція рідинного охолодження в стійкові системи вимагає координації між конструкцією, розподілом живлення та управлінням охолоджуючою рідиною.

Оскільки оператори розгортають кластери з потужними графічними процесорами для штучного інтелекту, теплова продуктивність на рівні чіпа стає пов'язаною з часом безвідмовної роботи та енергоефективністю.

Legrand описує придбання Kratos та партнерство Accelsius як два стовпи єдиної стратегії центрів обробки даних. Один з них зміцнює основу живлення «сірих» зон, а інший просуває охолодження наступного покоління у «білих» зонах.

Разом ці кроки дозволяють Legrand вирішувати критично важливі інфраструктурні вимоги, від розподільчих пристроїв середньої напруги до рідинного охолодження на рівні стійок.

3.2 Розробка структурної схеми

Коли бізнес зростає, повинна зростати і IT-інфраструктура. Однак ось у чому річ: без міцної кабельної основи таке зростання може зіткнутися з деякими суттєвими перешкодами. Саме тут на допомогу приходить масштабована структурована кабельна система. Незалежно від того, чи розширюєте ви офісний

					ВКРБ-123.26.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

простір, додаєте нові пристрої чи переходите до хмарних додатків, правильна установка мережевого кабелю гарантує ефективну роботу вашого бізнесу сьогодні *та* в майбутньому.

Більшість підприємств рано усвідомлюють важливість структурованої проводки, особливо враховуючи зростання попиту на швидший інтернет, безпечний зв'язок та операції з великою кількістю пристроїв. Проектування системи, орієнтованої на майбутнє, – це не просто протягування кабелів крізь стіни. Йдеться про розумне планування, вибір правильних компонентів та розуміння ролі масштабованості з першого дня. У цьому посібнику ми розповімо вам, як створити низьковольтну кабельну конструкцію, яка адаптується та розвивається разом з вашими потребами, забезпечуючи безперебійні та економічно ефективні рішення.

Думайте наперед: чому масштабованість важлива з самого початку

Масштабована система не є неов'язковою, вона є необхідною. Уявіть, що ви переїжджаєте в новий офіс, а потім виявляєте, що кабельне з'єднання не може підтримувати ваші нові VoIP-телефони або високошвидкісні сервери. Раптом ваша мережа починає працювати нестабільно, а продуктивність падає.

Ось чому планування майбутньої пропускної здатності, додаткових кінцевих точок та нових технологій з самого початку є докорінним зміною правил гри. Структурована кабельна система – це не лише поточна схема, а й перспективна інвестиція. Гнучкість, вбудована в масштабовану систему, дозволяє легко переміщувати, додавати та змінювати (MAC-адреси) без серйозних перебоїв або дорогої перемонтажі.

Передбачаючи зміни та плануючи зростання, ви готуєте свій бізнес до плавних переходів, меншої кількості проблем з обслуговуванням та нижчих довгострокових витрат.

Почніть з детальної оцінки ділянки

Неможливо спроектувати потужну структуровану кабельну систему без розуміння вашої поточної інфраструктури. Проведення ретельного обстеження

					ВКРБ-123.26.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

об'єкта допоможе вам спланувати фізичний простір, визначити точки доступу та оцінити існуюче обладнання.

Ви знаходитесь в окремому офісі чи в багатоповерховому приміщенні? Вам потрібно підключити зовнішні камери безпеки або точки доступу Wi-Fi? Все це впливає на те, як має бути прокладена та підтримувана ваша кабельна мережа. Враховуйте поточні та майбутні потреби в пропускній здатності. Більше співробітників або пристроїв означає більше кабельних терміналів і, можливо, модернізованих кабелів, таких як Cat6a або оптоволоконні.

Цей початковий крок також виявляє приховані обмеження, такі як обмеження відстані, перешкоди від системи опалення, вентиляції та кондиціонування повітря або фізичні перешкоди, які можуть вплинути на продуктивність або безпеку.

Виберіть правильні типи кабелів для довгострокового успіху

Вибір неправильних кабелів схожий на встановлення дешевих шин на гоночний автомобіль – деякий час це працюватиме, але продуктивність швидко постраждає. Для кабелів для передачі голосу та даних галузевим стандартом зазвичай є Cat5e, Cat6 або Cat6a, але правильний вибір залежить від ваших потреб у пропускній здатності та вимог до відстані.

Якщо ви очікуєте вищої швидкості передачі даних або використовуєте такі програми, як відеоконференції, оберіть кабель Cat6a або навіть оптоволоконний кабель. Ці варіанти підтримують вищі частоти та зменшують перехресні перешкоди. Оптоволоконний кабель може передавати дані на набагато більші відстані без втрати якості сигналу.

Не забудьте відокремити низьковольтні кабелі, такі як системи сигналізації або системи контролю доступу, від високовольтних ліній електропередач, щоб уникнути електричних перешкод. Надійний, структурований план електропроводки враховує окремі ролі кожного типу кабелю та інтегрує їх чисто та безпечно.

					ВКРБ-123.26.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

Використовуйте централізовані, організовані кабельні шляхи

Безлад із заплутаних кабелів може швидко перетворитися на технологічну катастрофу. Структурована кабельна система значною мірою залежить від чистої, організованої та централізованої системи управління кабелями. Подумайте про кабельні лотки, короба, патч-панелі та марковані кінцеві роз'єми.

Групуйте кабелі за призначенням та прокладайте їх позначеними шляхами, такими як стояки для вертикальної прокладки кабелів або стельові лотки для горизонтальних прокладок. Маркування є ключовим. Майбутні техніки (або ви самі) подякують вам за чітко позначені кінцеві точки та легко відстежувані лінії.

Правильна організація також забезпечує кращий потік повітря навколо мережевого обладнання, зменшуючи ризик перегріву та простоїв. Чисту кабельну систему не лише легше обслуговувати, вона також підвищує загальну надійність та довговічність вашої мережі.

Не ігноруйте галузеві стандарти та відповідність вимогам

Хоча кожен бізнес хоче індивідуального рішення, ваш проект кабельної мережі все одно повинен відповідати найкращим галузевим практикам, таким як стандарти TIA/EIA. Ці норми гарантують, що ваша мережева кабельна система буде безпечною та високопродуктивною.

Використання затверджених номіналів кабелів, дотримання обмежень щодо радіуса вигину та прокладання кабелів подалі від електромагнітних перешкод (EMI) – це лише кілька важливих рекомендацій. Відповідність вимогам виходить за рамки безпеки – вона гарантує безперебійну роботу вашої системи як із сучасними, так і з майбутніми технологіями.

Якщо ваш бізнес коли-небудь проходитиме IT-аудит або сертифікацію (наприклад, HIPAA або PCI), кабельні системи, що відповідають вимогам, вже забезпечать вам крок попереду.

					ВКРБ-123.26.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

Плануйте легке розширення та оновлення

Погляньмо правді в очі: ваша кабельна система не буде статичною. Зі зростанням вашого бізнесу зростатимуть і ваші потреби в пристроях, даних і пропускній здатності. Вбудовуйте варіанти розширення з самого початку.

Це означає включення додаткових каналів, запасних портів на патч-панелях та вільних місць для кабелів для майбутніх прокладок. Також розумно сегментувати мережу, наприклад, розділити системи голосового зв'язку та кабельних систем даних, щоб забезпечити модернізацію однієї, не впливаючи на іншу.

А коли поширення хмарних технологій зросте (що, ймовірно, станеться), ваша магістральна кабельна та бездротова інфраструктура повинні бути готові підтримувати підвищене навантаження. Проектування модульної системи сьогодні дозволяє уникнути повної перебудови в майбутньому.

Працюйте з досвідченими фахівцями зі структурованої кабельної системи

Незалежно від того, скільки досліджень ви проведете, проектування масштабованої системи найкраще довірити експертам. Шукайте компанії, які спеціалізуються на рішеннях для структурованої проводки та мають досвід роботи з різноманітними комерційними середовищами.

Професіонали не просто прокладають кабелі – вони аналізують вашу бізнес-модель, передбачають зростання та проектують системи, які адаптуються з часом. Правильний партнер також забезпечить вашу готовність до майбутніх технологій, таких як мережі зі швидкістю 10 Гбіт/с або інтегрована автоматизація будівель.

Перспективи майбутнього завдяки волоконній оптиці та передовим технологіям

Один із надійних способів зберегти масштабованість – це підготувати вашу інфраструктуру до оптоволоконного зв'язку. Хоча це може коштувати

					ВКРБ-123.26.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

дорожче на початку, переваги, включаючи вищу пропускну здатність і більшу надійність, значно переважають початкові витрати.

Поєднання вашої структурованої кабельної системи з новими технологіями, такими як живлення через Ethernet (PoE), інтелектуальне освітлення та системи контролю доступу, створює пов'язане середовище, готове до всього. Це не просто масштабованість – це інтелектуальне зростання.

На завершення, проектування масштабованої структурованої кабельної системи є важливим для забезпечення майбутнього вашої IT-інфраструктури. Плануючи зростання, дотримуючись стандартів та використовуючи високоякісні компоненти, ви можете гарантувати, що ваша мережа підтримуватиме зростаючі вимоги до пропускну здатності та нові технології. Масштабованість мінімізує час простою, зменшує довгострокові витрати та дозволяє безперешкодне оновлення. Незалежно від того, чи ви будете систему з нуля, чи модернізуєте існуючу, співпраця з професіоналами забезпечує оптимальну продуктивність та адаптивність.

Структурована кабельна система планується й будується ієрархічно, з головною магістраллю й численними відгалуженнями від її.

Типова ієрархічна структура структурованої кабельної системи (рисунок 3.1) включає:

- горизонтальні кабельні підсистеми (у межах поверху);
- вертикальні кабельні підсистеми (усередині будівлі);
- магістральну кабельну підсистему ЦОД (у межах однієї території з декількома будинками);
- кабельну підсистему робочих місць;
- центральні комутаційні вузли будівель.

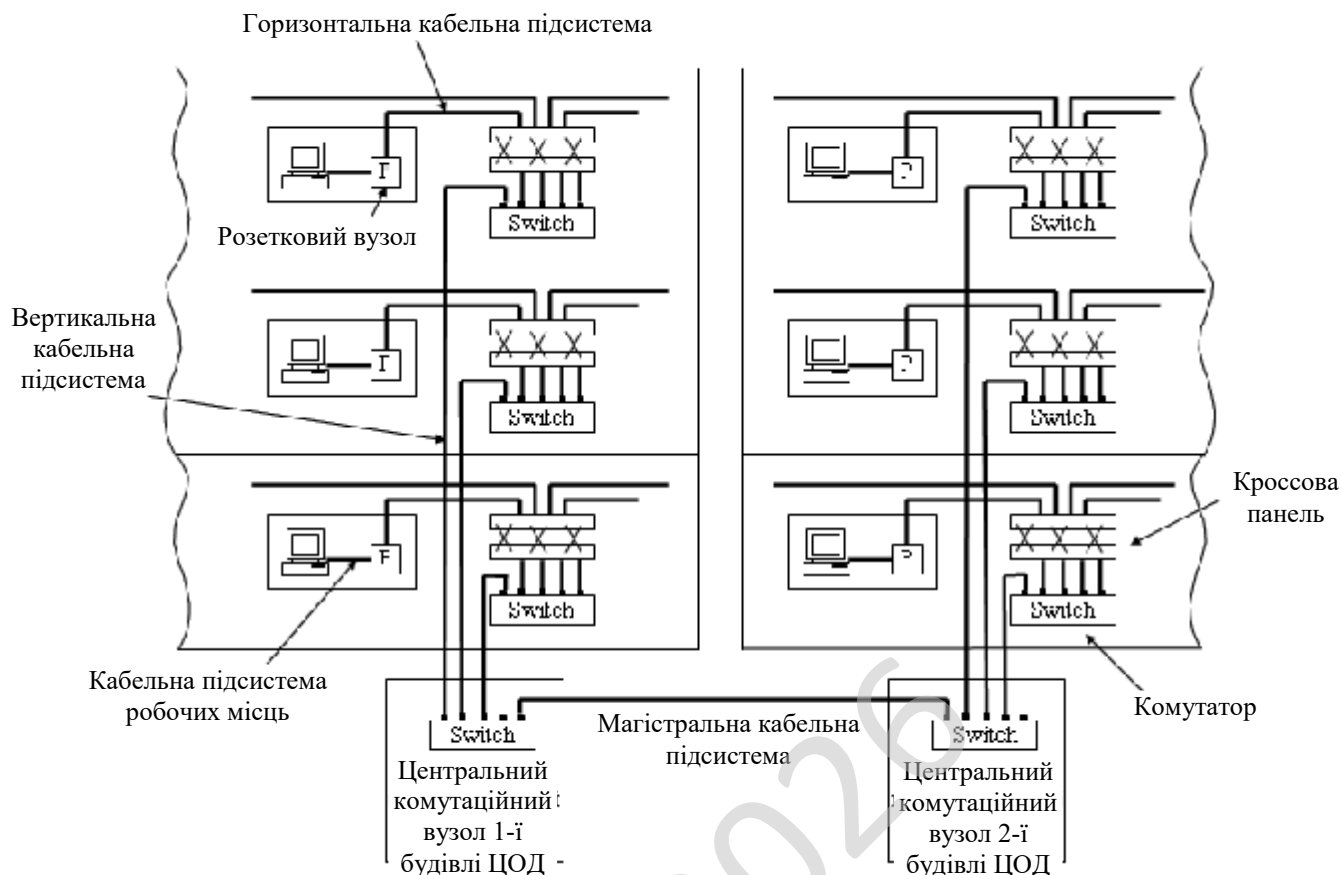


Рисунок 3.1 – Структурна схема системи

Стратегічна база для стратегії структурної кабельної системи

Стратегія структурної кабельної мережі 2026 закладає основу для того, як організації повинні планувати фізичну інфраструктуру в умовах зростання цифрового попиту. Оскільки підприємства прагнуть до більшої пропускної здатності та меншої затримки, рішення щодо кабельної мережі тепер впливають на довгострокову продуктивність та контроль витрат. Тому керівники повинні узгоджувати вибір фізичного рівня із зростанням застосунків та впровадженням хмарних технологій. Крім того, рання стандартизація зменшує майбутні переробки та операційні труднощі.

Стратегія планування потужностей та структурної кабельної системи

Стратегія структурної кабельної мережі 2026 надає пріоритет плануванню потужності над реактивними оновленнями. Хоча багато середовищ все ще

покладаються на застарілі мідні кабелі, вимоги до пропускної здатності продовжують зростати. Як результат, категорія 6A залишається мінімальною базовою для нових проєктів. Водночас впровадження багатомодового оптоволоконного волокна розширюється на всіх рівнях доступу. Отже, організації отримують гнучкість для майбутніх комутаційних платформ та розширення бездротового зв'язку. Крім того, структуровані шляхи та належне маркування покращують операційну видимість та пришвидшують усунення несправностей. Завдяки цьому команди можуть швидше масштабуватися без створення ризиків.

Стратегія центрів обробки даних та структурної кабельної системи

Стратегія структурної кабельної мережі 2026 року змінює дизайн центрів обробки даних з урахуванням щільності та потоку повітря. Хоча площа обладнання зменшується, кількість портів продовжує зростати. Тому рішення з високою щільністю оптоволоконного волокна стають важливими. Крім того, модульні кабельні системи скорочують час розгортання та підтримують поетапне зростання. Оскільки робочі навантаження переходять між локальними та хмарними ресурсами, фізична інфраструктура повинна залишатися адаптивною. Навпаки, жорсткі конструкції обмежують гнучкість та збільшують витрати на життєвий цикл. З цієї причини стандартизовані кабельні архітектури забезпечують вимірну бізнес-цінність.

Стратегія відповідності та структурної кабельної системи

Стратегія структурної кабельної системи 2026 також посилює дисципліну щодо дотримання вимог та документації. Чіткі записи підтримують аудити, перевірки безпеки та безперервність роботи. Тим часом, послідовне тестування та сертифікація захищають базові показники продуктивності. Оскільки регуляторний тиск продовжує зростати, відповідність фізичному рівню більше не можна ігнорувати. Зрештою, організації, які ставляться до кабельних систем як до стратегічної інфраструктури, отримують стійкість, масштабованість та передбачувану продуктивність. Зі прискоренням технологічних циклів

					ВКРБ-123.26.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

дисциплінована стратегія кабельної системи стає конкурентною перевагою, а не безповоротними витратами.

Підсумовуючи, перспективні рішення щодо кабельних систем сприяють зростанню, знижують ризики та захищають капітальні інвестиції, оскільки підприємства рухаються ближче до 2026 року та далі.

3.3 Розробка функціональної схеми

Функціональна схема системи зображена на рисунку 3.2. З рисунку видно, що розроблена система складається з наступних блоків:

- Статистика подій системи Legrand Cabling System 3 для ЦОД.
- Робота з ресурсами системи Legrand Cabling System 3 для ЦОД.
- Робота з сесіями системи Legrand Cabling System 3 для ЦОД.
- Функції для роботи з центрами обробки даних, побудованих з використанням системи Legrand Cabling System 3 для ЦОД.
- Моніторинг трафіку системи Legrand Cabling System 3 для ЦОД.
- Робота з файлами системи Legrand Cabling System 3 для ЦОД.
- Монітор з'єднань системи Legrand Cabling System 3 для ЦОД.

Розглянемо детальніше кожний з блоків.

Робота з ресурсами системи Legrand Cabling System 3 для ЦОД включає в себе:

- Визначення доступних ресурсів ЦОД.
- Закриття локального ресурсу ЦОД.
- Відкриття локального ресурсу ЦОД.
- Приховання й показ ресурсів ЦОД.

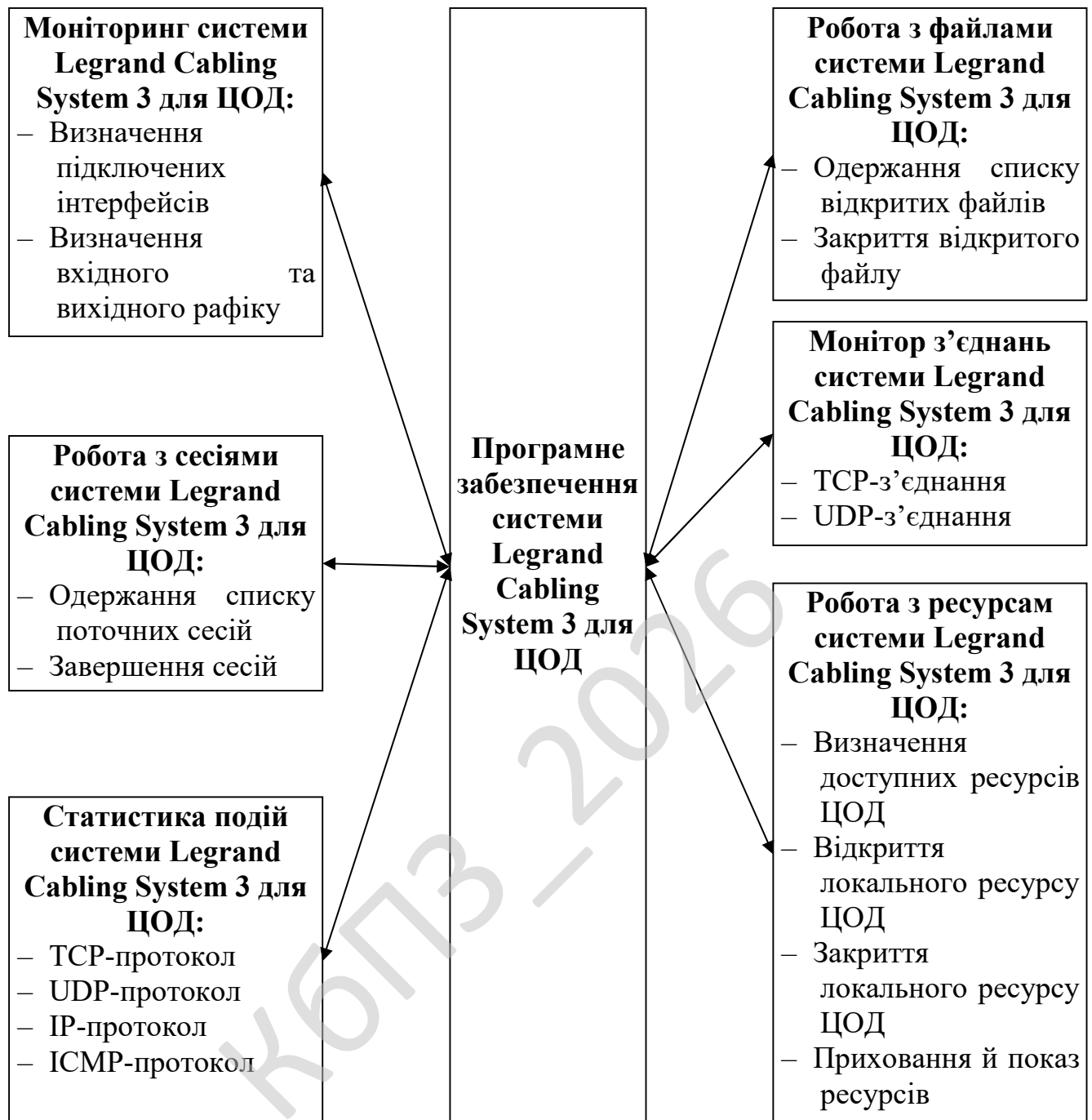


Рисунок 3.2 – Функціональна схема системи

Система відображає наявні у мережі ресурси у вигляді дерева. Пошук ресурсів можна здійснювати по заданим умовам: локальні чи глобальні ресурси; всі ресурси, тільки файли, чи тільки принтери, тощо.

Можна додавати до загальних ресурсів мережі свої власні, а також закривати їх потім.

Робота з сесіями системи Legrand Cabling System 3 для ЦОД включає в себе:

- Одержання списку поточних сесій.
- Завершення сесій.

Програма дозволяє переглянути список відкритих сесій, що включає в себе: назву сесії, користувача, що її розпочав, номер сесії, час роботи та час очікування.

Моніторинг трафіку системи Legrand Cabling System 3 для ЦОД включає в себе:

- Визначення підключених інтерфейсів.
- Визначення вхідного та вихідного трафіку.

Розроблена система відображає всі інтерфейси приєднані до комп'ютера, на якому запущена програма, їх MAC-адреси та вхідний і вихідний трафік на кожному з них.

Робота з файлами системи Legrand Cabling System 3 для ЦОД включає в себе:

- Одержання списку відкритих файлів.
- Закриття відкритого файлу.

Можна переглянути, які з Ваших файлів, що Ви відкрили для загального доступу, переглядають по мережі. Програма відобразить список файлів та користувачів, які їх переглядають. Також можна відкрити чи закрити файл.

Монітор з'єднань системи Legrand Cabling System 3 для ЦОД включає в себе:

- Відстеження TCP- з'єднань.
- Відстеження UDP- з'єднань.

Система фіксує всі підключення по TCP- та UDP-протоколу, та виводить їх на екран у форматі *IP-адреса:порт_призначення*.

Статистика подій системи Legrand Cabling System 3 для ЦОД включає в себе відстеження подій в наступних протоколах:

					ВКРБ-123.26.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

- TCP-протокол.
- UDP-протокол.
- IP-протокол.
- ICMP-протокол.

Статистика ведеться по цілому ряду параметрів. Наприклад для TCP-протоколу фіксується: Тип алгоритму повторної передачі, мінімальний тайм-аут, максимальний тайм-аут, максимальна кількість помилок з'єднання, активні з'єднання, пасивні з'єднання, невдалі спроби відкриття, скидання встановлених з'єднань, отримані сегменти, надіслані сегменти, повторно передані сегменти, помилки тощо.

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

3.4 Розробка діаграми процесів

Розглянемо розроблену діаграму процесів яка зображена на рисунку 3.3. Основна будова діаграми процесів полягає у графічному представленні складу сукупностей даних, що характеризуються як співвідношення різних частин кожної з сукупностей.

Склад статистичної сукупності графічно може бути представлений як за допомогою абсолютних, так і відносних показників. Графічне зображення складу сукупності по абсолютними і відносними показниками сприяє проведенню більш глибокого аналізу і дозволяє проводити аналіз системи.

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.

При детальному її розгляді можна побачити як саме проходить взаємодія у розробленій системі. Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі.

Діаграми потоків даних містять чотири типи елементів:

- Зовнішні по відношенню до системи сутності.
- Потоки даних між елементами трьох попередніх типів.
- Процеси які являють собою трансформацію даних в рамках описуваної системи.
- Сховища даних (репозиторії).

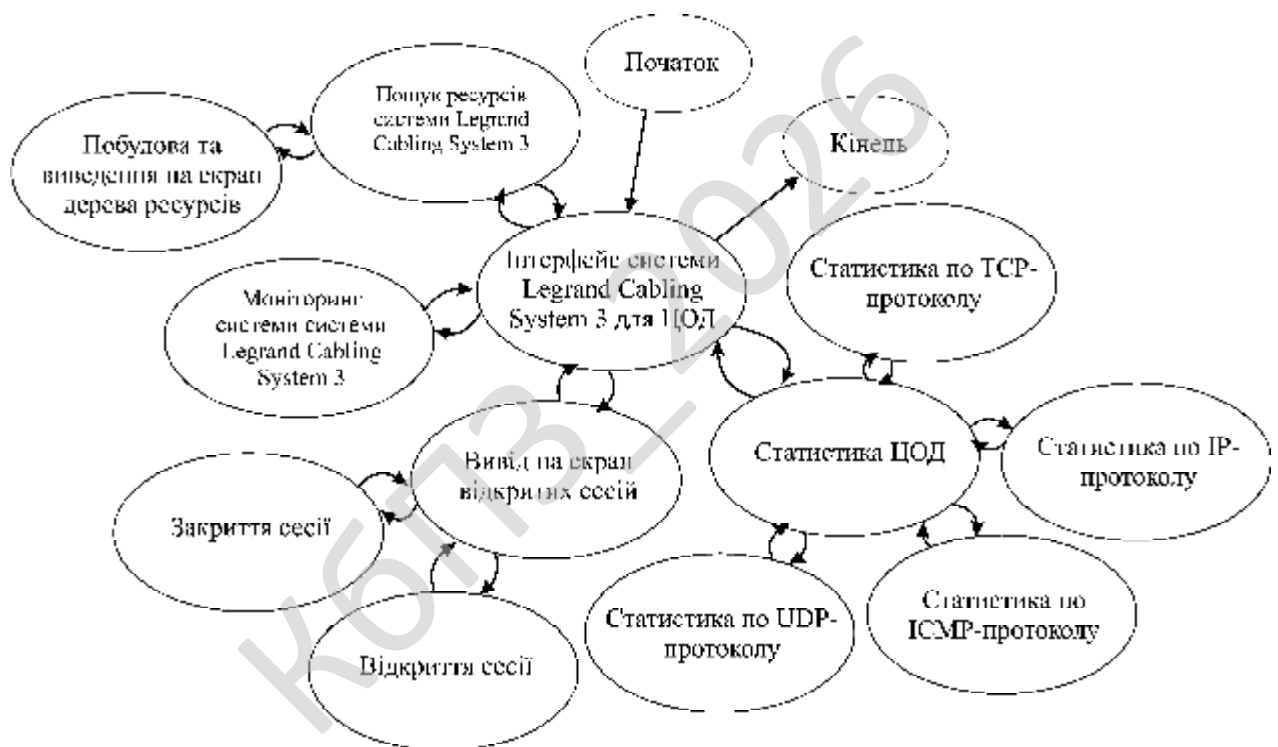


Рисунок 3.3 – Діаграма взаємодії процесів

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.

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

Розглянемо реалізацію бакалаврської дипломної роботи. Були проведені розрахунки і підібрані набори тестових даних для перевірки правильності реалізації проектних рішень.

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 підсистеми та останньої стадії – перевірка поточного стану з завершенням роботи розробленого ПЗ.

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

Розглянемо формат що використовується – **JSON** (JavaScript Object Notation, укр. запис об'єктів JavaScript, вимовляється джейсон) – це текстовий формат обміну даними між комп'ютерами.

JSON базується на тексті, може бути прочитаним людиною. Формат дозволяє описувати об'єкти та інші структури даних.

Цей формат головним чином використовується для передачі структурованої інформації через мережу (завдяки процесу, що називають серіалізацією). Розробив і популяризував формат Дуглас Крокфорд.

JSON знайшов своє головне призначення у написанні веб-програм, а саме при використанні технології AJAX. JSON, що використовується в AJAX, виступає як заміна XML (використовується в AJAX) під час асинхронної передачі структурованої інформації між клієнтом та сервером.

При цьому перевагою JSON перед XML є те, що він дозволяє складні структури в атрибутах, займає менше місця і прямо інтерпретується за допомогою JavaScript в об'єкти.

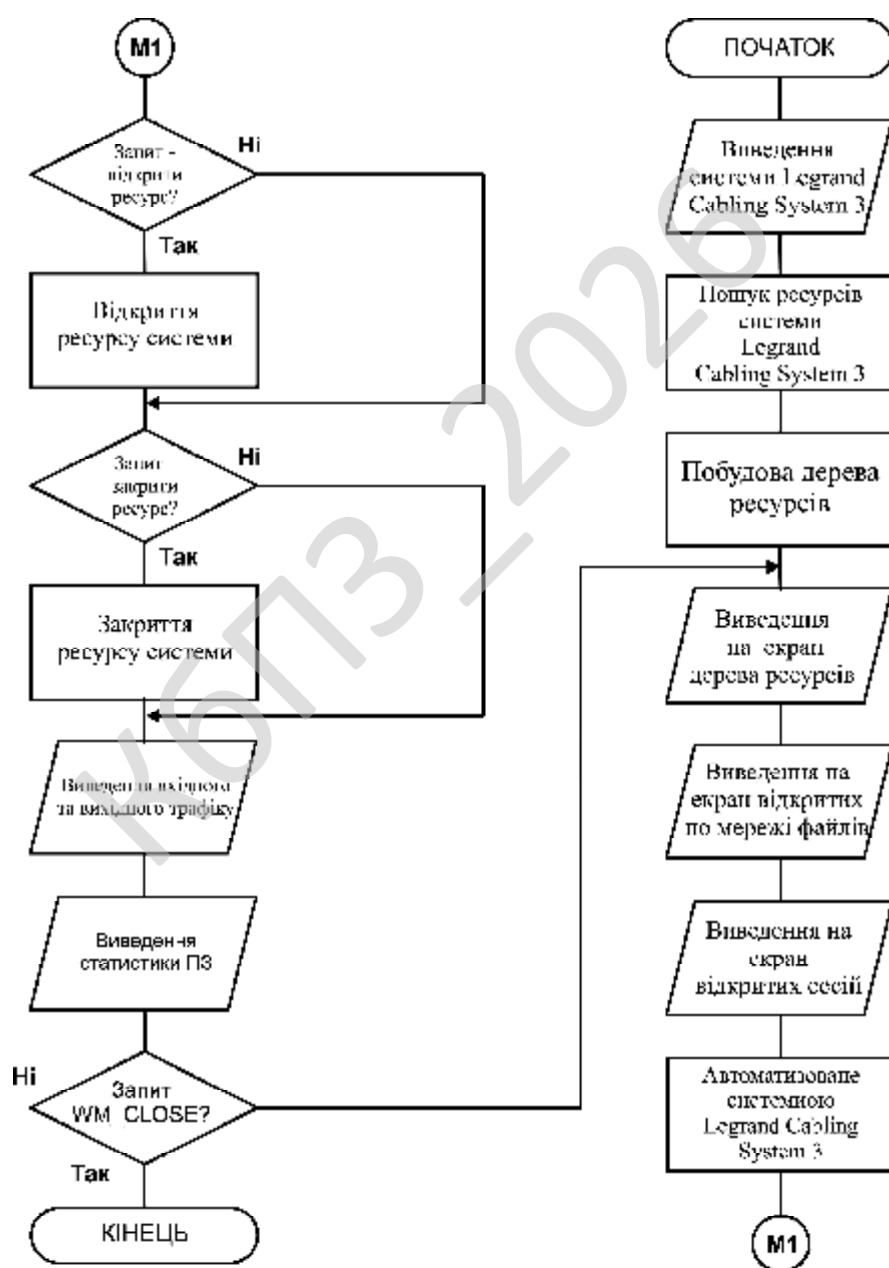


Рисунок 4.1 – Блок-схема основної програми

JSON з'явився через необхідність обміну даними з сервером у реальному часі без використання плагінів для браузерів, flash-додатків або Java-апплетів, які використовувались скрізь на початку 2000-х років.

Дуглас Крокфорд був тим, хто активно просував новий на той час формат. Він з колегами хотів створити технологію, яка використовувала б можливості звичайного браузера давала б веб-розробникам можливість створювати веб-додатки із постійним двостороннім зв'язком із веб-сервером.

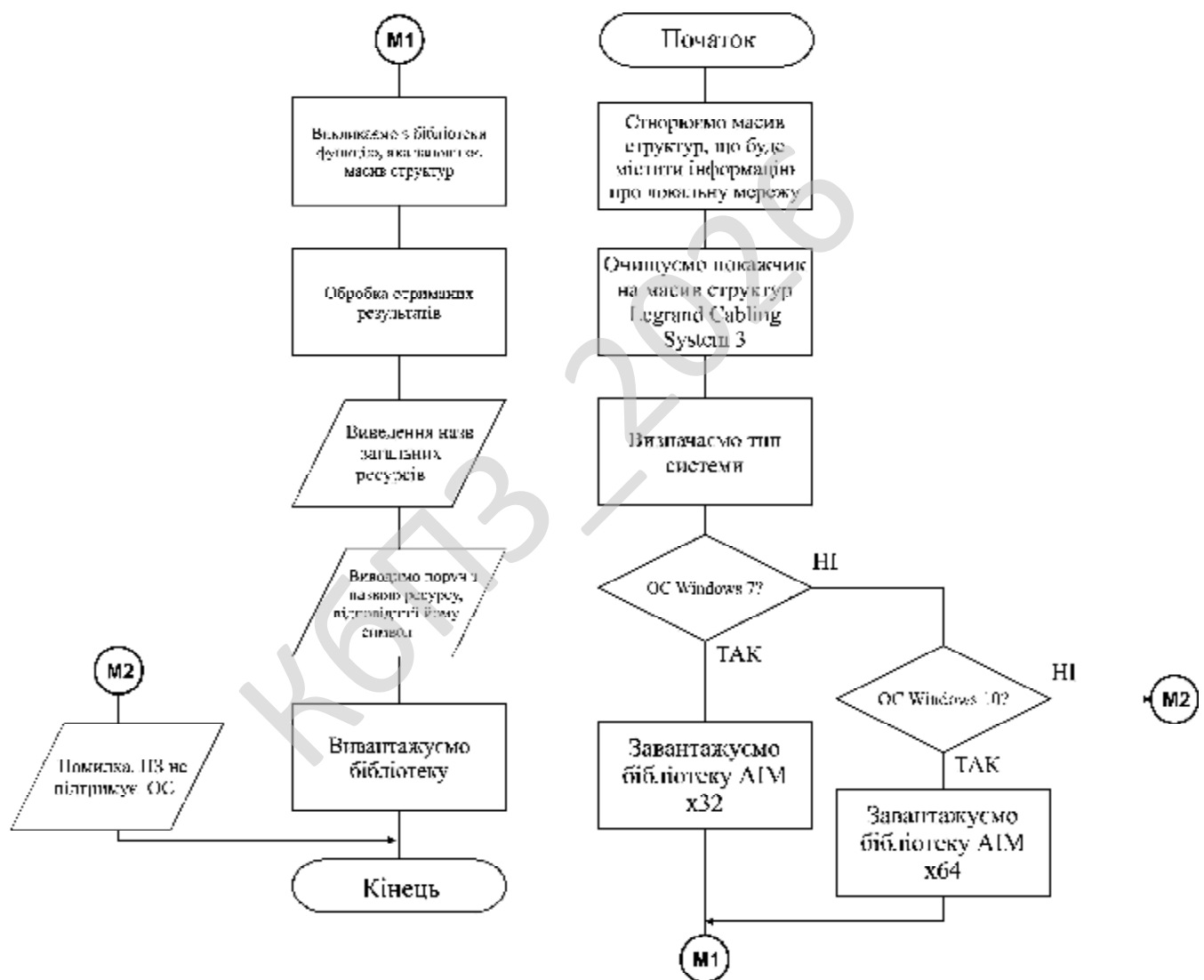


Рисунок 4.2 – Блок-схема роботи підпрограми

JSON вперше був використаний в проєкті в Communities.com для Cartoon Network, він дозволяв обмінюватись повідомленнями і одночасно маніпулювати DHTML-елементами.

Веб-сайт JSON.org було запущено 2002 року. У грудні 2005-го року Yahoo! почав переводити деякі зі своїх веб-сервісів на роботу з JSON. Google взялася до роботи з технологією у своєму веб-протоколі GData у грудні 2006-го року.

Використання

За рахунок своєї лаконічності в порівнянні з XML, формат JSON може бути більш придатним для серіалізації складних структур.

Якщо говорити про веб-застосунки, в такому ключі він доречний в задачах обміну даними як між браузером і сервером (AJAX), так і між самими серверами (програмні HTTP-інтерфейси).

Формат JSON так само добре підходить для зберігання складних динамічних структур в реляційних базах даних або файловому кеші.

Приклад використання JSON (JavaScript)

```
const ajaxObj = JSON.parse(ajaxData)
console.log(`${ajaxObj.name}_${ajaxObj.rates[2]}`)
```

Синтаксис

JSON будується на двох структурах:

1. Набір пар назва/значення. У різних мовах це реалізовано як об'єкт, запис, структура, словник, хеш-таблиця, список з ключем або асоціативним масивом.

2. Впорядкований список значень. У багатьох мовах це реалізовано як масив, вектор, список, або послідовність.

Це універсальні структури даних. Теоретично всі сучасні мови програмування підтримують їх у тій чи іншій формі.

Оскільки JSON використовується для обміну даними між різними мовами програмування, то є сенс будувати його на цих структурах.

У JSON використовуються такі їхні форми:

					ВКРБ-123.26.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

1. Об'єкт – це послідовність пар назва/значення. Об'єкт починається з символу { і закінчується символом } . Кожне значення слідує за : і пари назва/значення відділяються комами.

2. Масив – це послідовність значень. Масив починається символом [і закінчується символом]. Значення відділяються комами.

3. Значення може бути рядком в подвійних лапках, або числом, або логічними true чи false, або null, або об'єктом, або масивом. Ці структури можуть бути вкладені одна в одну.

4. Рядок – це послідовність з нуля або більше символів юнікода, обмежена подвійними лапками, з використанням escape-послідовностей, що починаються зі зворотної косої риски \. Символи представляються простим рядком. Тип Рядок (String) дуже схожий на String в мовах C і Java.

Число теж дуже схоже на C- або Java-число, за винятком того, що вісімкові та шістнадцяткові формати не використовуються. Пропуски можуть бути вставлені між будь-якими двома лексемами.

Наступний приклад показує JSON представлення об'єкта, що описує людину. У об'єкті є рядкові поля імені і прізвища, об'єкт, що описує адресу, і масив, що містить список телефонів.

```
{  "firstName": "Іван",  
    "phoneNumbers": [  
        "044 123-1234",  
        "050 123-4567"    ] }
```

Використання JSON в AJAX

Наступний фрагмент коду JavaScript показує, як клієнт може використати XMLHttpRequest для запиту об'єктів в форматі JSON з серверу. (Серверна частина коду пропущена, вона просто повертає на запит URL рядок у JSON форматі).

Треба відзначити, що тут використання XMLHttpRequest не є кросс-браузерним (за деталями звертайтеся на сторінку XMLHttpRequest), і код зазнаватиме незначних модифікацій в різних версіях оглядачів Internet Explorer, Opera, Safari або Mozilla. Використання запиту XMLHttpRequest обмежено

правилом одного джерела (same origin policy): URL, що відповідає на запит, має посилатися на той же сайт, що обслуговує сторінку, що ініціювала запит.

Оглядачі можуть також використовувати тег `<iframe>` для асинхронного запиту JSON-даних в кросс-браузерному варіанті, або використати просте перенаправлення `<form action="url_to_cgi_script" target="name_of_hidden_iframe">`. Такий підхід був поширений до приходу популярного нині запиту XMLHttpRequest.

Динамічний тег `<script>` також можна використати для підвантаження JSON-даних. Ця техніка можлива, щоб обійти надто суворе правило одного джерела, але вона не є безпечною. Запит XMLHttpRequest пропонується, як безпечніша альтернатива.

Питання безпеки

Хоча JSON призначений для передачі даних в серіалізованому вигляді, його синтаксис відповідає синтаксису JavaScript і це створює низку проблем безпеки. Часто для обробки даних, отриманих від зовнішнього джерела у форматі JSON, до них застосовується функція `eval()` без якої-небудь попередньої перевірки.

JavaScript eval()

Оскільки JSON представляється синтаксично правильним фрагментом коду JavaScript, природним способом розбору JSON-даних в JavaScript-програмі є використання вбудованої в JavaScript функції `eval()`, яка призначена для обчислення JavaScript-виразів. При цьому підході відпадає необхідність у використанні додаткових парсерів.

Техніка використання `eval()` робить систему вразливою, якщо джерело JSON-даних, що використовуються, не відноситься до надійних. Такими даними може виступати шкідливий JavaScript-код для атак за допомогою ін'єкції коду. За допомогою цієї вразливості можливо здійснювати крадіжку даних, підробку автентифікації. Проте, вразливість можна усунути за рахунок використання додаткових засобів перевірки даних на коректність. Наприклад, до виконання

					ВКРБ-123.26.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

eval() отримані від зовнішнього джерела дані можуть перевірятися за допомогою регулярних виразів. У RFC, що визначає JSON пропонується використовувати такий код для перевірки його відповідності формату JSON

```
const my_JSON_object = !(/[^\s,:{}\\[\]0-9.\-+Eaeflnr-u \n\r\t]/.test(
text.replace(/"(\\"|.)*"/g, ''))) &&
eval('(' + testedData + ')');
```

Як безпечніша альтернатива eval() була запропонована нова функція parseJSON(), здатна обробляти тільки JSON-дані.

Вбудований JSON

Останні версії веб-браузерів мають вбудовану підтримку JSON і здатні його обробляти без виклику функції eval(), що призводить до описаної проблеми. Обробка JSON у такому разі зазвичай здійснюється швидше. Так у червні 2009 року вбудовану підтримку JSON мали такі браузери: Mozilla Firefox 3.5+; SeaMonkey 2; Thunderbird 3; Microsoft Internet Explorer 8; Opera 10.5; Браузери, засновані на WebKit (наприклад, Google Chrome, Apple Safari).

Принаймні дві популярні бібліотеки JavaScript використовують вбудований JSON у разі його доступності: jQuery; Dojo.

Підrobка крос-доменного запиту

Непродумане використання JSON робить сайти вразливими до підrobки міжсайтових запитів (CSRF або XSRF). Оскільки тег <script> допускає використання джерела, що не належить до того ж домену, що і використовуваний ресурс, це дозволяє виконувати код даних, представлених у форматі JSON, в контексті довільної сторінки, що робить можливою компрометацію паролів або іншої конфіденційної інформації користувачів, що пройшли авторизацію на іншому сайті.

Це є проблемою тільки у разі вмісту в JSON-даних конфіденційної інформації, яка може бути компрометована третьою стороною і якщо сервер розраховує на політику одного джерела, блокуючи доступ до даних при виявленні зовнішнього запиту.

					ВКРБ-123.26.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

Це не є проблемою, якщо сервер визначає допустимість запиту, надаючи дані тільки у разі його коректності. HTTP cookie не можна використовувати для визначення цього. Виключне використання HTTP cookie використовується підрубкою міжсайтових запитів.

Розширення, JSONP

JSONP або «JSON з підкладкою» є розширенням JSON, коли назва функції зворотного виклику вказується як вхідний аргумент. Спочатку ідея була запропонована в блозі MacPython в 2005 році, і в наш час використовується багатьма Web 2.0 застосунками, такими, як Dojo Toolkit Applications, Google Toolkit Applications і zanox Web Services.

Подальші розширення цього протоколу були запропоновані з урахуванням введення додаткових аргументів, як, наприклад, у разі JSONPP за підтримки S3DB вебсервісів.

Оскільки JSONP використовує скрипт-теги, виклики по суті відкриті світові. З цієї причини JSONP може бути недоречними для зберігання конфіденційних даних.

Включення скриптових тегів від віддалених сайтів дозволяє їм передати будь-який контент на сайті. Якщо віддалений сайт має вразливості, які дозволяють виконати ін'єкції JavaScript, то початковий сайт також може бути ними зачеплений.

Розширення, BSON

BSON – це бінарна форма представлення простих структур даних і асоціативних масивів (які називають об'єктами або документами). Назва «BSON» заснована на визначенні JSON і неофіційно означає «Binary JSON» (бінарний JSON).

JSON Reference

Стандарт JSON не описує посилання на інші об'єкти або частини, але існує чернетка стандарту IETF для посилань на об'єкти на основі JSON. Посилання дозволяють здійснювати трансклюзію - вставляти одні документи в інші.

JSON Reference - це JSON-об'єкт з ключем "\$ref" (всі інші ключі ігноруються) і значенням стрічкового типу що містить URI, наприклад:

```
{ "$ref": "http://example.com/example.json#/foo/bar" }
```

Якщо URI містить ідентифікатор фрагмента (в прикладі вище "/foo/bar"), він інтерпретується як JSON Pointer.

Модуль dojoy.json.ref в Dojo toolkit, забезпечує підтримку декількох форм JSON Reference.

Розглянемо протокол SNMP (Simple Network Management Protocol, простий протокол керування мережею) – це протокол керування мережами зв'язку на основі архітектури TCP/IP.

На основі концепції TMN в 1980–1990 р. різними органами стандартизації був вироблений ряд протоколів керування мережами передачі даних з різним спектром реалізації функцій TMN. До одного з типів таких протоколів керування належить Simple Network Management Protocol (SNMP).

SNMP – це технологія, покликана забезпечити керування й контроль за пристроями й програмами в мережі зв'язку шляхом обміну керуючою інформацією між агентами, що розташовуються на мережних пристроях, і менеджерами, розташованими на станціях керування. SNMP визначає мережу як сукупність мережних керуючих станцій й елементів мережі (головні машини, шлюзи й маршрутизатори, термінальні сервери), які спільно забезпечують адміністративні зв'язки між мережними керуючими станціями й мережними агентами. SNMP різних версій присвячений цілий ряд рекомендацій IETF (RFC).

Зазвичай при використанні SNMP присутні керовані та керуючі системи. До складу керованої системи входить компонент, який називається агентом, який відправляє звіти керуючій системі. По суті SNMP агенти передають управлінську інформацію на керуючі системи як змінні (такі як «вільна пам'ять», «ім'я системи», «кількість працюючих процесів» тощо).

Керуюча система може отримати достовірну інформацію через операції протоколу GET, GETNEXT і GETBULK. Агент може самостійно без запиту надсилати дані, використовуючи операцію протоколу TRAP або INFORM.

					ВКРБ-123.26.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

Управляючі системи можуть також відправляти конфігураційні оновлення або контролюючі запити, використовуючи операцію SET для безпосереднього управління системою. Операції конфігурування та управління використовуються тільки тоді, коли потрібні зміни у мережній інфраструктурі. Операції моніторингу зазвичай виконуються на регулярній основі.

Змінні, доступні через SNMP, організовані в ієрархії. Ці ієрархії та інші метадані (такі як тип і опис змінної) описуються Базами Керуючої Інформації' (Management Information Bases (MIBs)).

SNMP не визначає, яку інформацію (які змінні) керована система повинна надавати. Навпаки, SNMP використовує розширювану модель, в якій доступна інформація визначається Базами Керуючої Інформації (MIB – Management Information Base).

Бази Керуючої Інформації описують структуру керуючої інформації пристроїв. Вони використовують ієрархічний адресний простір імен, що містить унікальний ідентифікатор об'єкта (object identifier (OID)).

Грубо кажучи, кожен унікальний ідентифікатор об'єкта ідентифікує змінну, яка може бути прочитана чи встановлена через SNMP. MIB'и використовують нотацію, визначену в ASN.1.

Ієрархія MIB може бути зображена як дерево з безіменним коренем, рівні якого приписані різними організаціями. На найвищому рівні MIB OID'и належать різним організаціям, що займаються стандартизацією, в той час як на нижчих рівнях OID'и виділяються асоційованими організаціями.

Ця модель забезпечує управління на всіх шарах мережної моделі OSI, адже MIB'и можуть бути визначені для будь-яких типів даних і операцій.

Керований об'єкт – це одна з будь-якого числа характеристик, специфічних для керованого пристрою. Керований об'єкт включає в себе один або більше екземплярів об'єкта (що ідентифікуються за OID), які насправді є змінними.

Існує два типи керованих об'єктів: Скалярні об'єкти визначають єдиний екземпляр об'єкта; Табличні об'єкти визначають множинні, пов'язані екземпляри об'єктів які групуються в таблицях MIB.

Прикладом керованого об'єкта може бути atInput, який є скалярним об'єктом що містить єдиний екземпляр об'єкта, ціле число, яке показує загальну кількість вхідних пакетів AppleTalk на мережний інтерфейс маршрутизатора.

Ідентифікатор об'єкта (OID) унікально ідентифікує керований об'єкт в ієрархії MIB.

В телекомунікаціях і комп'ютерних мережах, ASN.1 є стандартною гнучкою нотацією для опису структур даних, що служать для кодування, передачі і декодування даних. ASN.1 являє собою набір правил для опису структури об'єктів, незалежних від специфічних для обладнання методик кодування, і формальну нотацію, яка дозволяє уникнути неоднозначностей.

ASN.1 це єдиний ISO та ITU-T стандарт, спочатку визначений у 1984 році як частина стандарту CCITT X.409: 1984. Пізніше, в 1988 році, завдяки його широкому застосуванню, він був виділений в окремий стандарт X.208. Значно переглянута версія 1995року описана в X.680.

Адаптована підмножина ASN.1 Структура Керуючої Інформації (SMI) описана в протоколі SNMP для визначення наборів пов'язаних MIB об'єктів, також званих MIB модулями.

SNMP працює на прикладному рівні TCP/IP (сьомий рівень моделі OSI). Агент SNMP отримує запити по UDP-порту 161.

Менеджер може посилати запити з будь-якого доступного порту джерела на порт агента. Відповідь агента буде відправлений назад на порт джерела на менеджери.

Менеджер отримує повідомлення (Traps і InformRequests) по порту 162. Агент може генерувати повідомлення з будь-якого доступного порту. При використанні TLS або DTLS запити виходять по порту 10161, а пастки відправляються на порт 10162.

					ВКРБ-123.26.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

У SNMPv1 зазначено п'ять основних протокольних одиниць обміну (protocol data units - PDU). Ще дві PDU, GetBulkRequest і InformRequest, були введені в SNMPv2 і перенесені в SNMPv3.

Нижче перераховані сім протокольних одиниць обміну SNMP:

1. GetRequest. Запит від менеджера до об'єкту для отримання значення змінної або списку змінних. Необхідні змінні вказуються в полі variable bindings (розділ поля values при цьому не використовується). Отримання значень зазначеної змінної повинно бути виконано агентом як Атомарна операція. Менеджеру буде повернений Response (відповідь) з поточними значеннями.

2. SetRequest. Запит від менеджера до об'єкту для зміни змінної або списку змінних. Зв'язані змінні вказуються в тілі запиту. Зміни всіх зазначених змінних повинні бути виконані агентом як атомарна операція. Менеджеру буде повернений Response з (поточними) новими значеннями змінних.

3. GetNextRequest. Запит від менеджера до об'єкту для виявлення доступних змінних і їх значень. Менеджеру буде повернений Response зі зв'язаними змінними для змінної, яка є наступною в базі MIB в лексикографічному порядку. Обхід всієї бази MIB агента може бути проведений ітераційним використанням GetNextRequest, починаючи з OID 0. Рядки таблиці можуть бути прочитані, якщо вказати в запиті OID-и колонок в пов'язаних змінних.

4. GetBulkRequest. Покращена версія GetNextRequest. Запит від менеджера до об'єкту для численних ітерацій GetNextRequest. Менеджеру буде повернений Response з декількома пов'язаними змінними, обійденими починаючи зі пов'язаної змінної (змінних) в запиті. Специфічні для PDU поля non-repeaters і max-repetitions використовуються для контролю за поведінкою відповіді. GetBulkRequest був введений в SNMPv2.

5. Response. Повертає зв'язані змінні і значення від агента менеджеру для GetRequest, SetRequest, GetNextRequest, GetBulkRequest і InformRequest. Повідомлення про помилки забезпечуються полями статусу помилки і індексу помилки.

					ВКРБ-123.26.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

Ця одиниця використовується як відповідь і на Get-, і на Set-запити, в SNMPv1 називається GetResponse.

6. Trap. Асинхронне повідомлення від агента - менеджера. Включає в себе поточне значення sysUpTime, OID, що визначає тип trap (пастки), і необов'язкові зв'язані змінні. Адресація одержувача для пасток визначається за допомогою змінних trap-конфігурації в базі MIB. Формат trap-повідомлення був змінений в SNMPv2 і PDU перейменували в SNMPv2-Trap.

7. InformRequest. Асинхронне повідомлення від менеджера менеджера або від агента менеджера. Повідомлення від менеджера менеджера були можливі вже в SNMPv1 (за допомогою Trap), але SNMP зазвичай працює на протоколі UDP, в якому доставка повідомлень не гарантована, і не повідомляється про втрачені пакетах. InformRequest виправляє це зворотним відправленням підтвердження про отримання. Одержувач відповідає Response-му, що повторює всю інформацію з InformRequest. Цей PDU був введений в SNMPv2.

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.

					ВКРБ-123.26.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

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

					ВКРБ-123.26.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

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.

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, Delphi.

					ВКРБ-123.26.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

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).

4.2 Захист розробленого програмного забезпечення

Дані які використовуються у даній роботі захищаються алгоритмом ДСТУ 7564:2014 («Купина»). В Україні на основі консервативного підходу із залученням відомих і добре досліджених конструкцій була розроблена геш-функція, що базується на новому блоковому шифрі „Калина” (ДСТУ 7624:2014).

Національний стандарт ДСТУ 7564:2014 визначає криптографічну функцію гешування „Купина”, додатковий режим її застосування для формування коду автентифікації повідомлення (імітовставки), а також значення для перевірки реалізацій.

Для скорочення обсягу тексту національного стандарту була застосована математична нотація, що дозволяє отримати точний і компактний запис.

Водночас, такий підхід може ускладнювати сприйняття сутності алгоритму для фахівців, що не мають фундаментальної криптологічної освіти.

У роботі наводиться розгорнутий альтернативний опис функції гешування „Купина” із позначеннями, традиційними для галузі комп’ютерних наук.

Термінологія та позначення

Вектор ініціалізації – бітова послідовність фіксованої довжини (512 або 1024 біта), що використовується як початкове значення при обчисленні геш-значення.

Внутрішній стан – бітова послідовність фіксованої довжини (512 або 1024 біта), що є проміжним значенням на кожній ітерації перетворення функції

гешування, а також вхідним та вихідним значеннями перетворень P і Q ; для цих перетворень внутрішній стан подається як матриця розміром $8 \times c$ байт.

Геш-значення (геш-вектор) – бітова послідовність фіксованої довжини ($n = 8 \cdot s, s \in \{1, 2, \dots, 64\}$), що є результатом роботи функції гешування.

Доповнення – вставка додаткових біт у кінець повідомлення для отримання кратності довжини бітової послідовності довжині внутрішнього стану функції гешування.

Повідомлення – бітова послідовність довжини від 0 біт (порожній рядок) до $2^{96} - 1$ біт.

Функція стиснення – ітеративне перетворення, що відображає l -бітний блок повідомлення та l -бітне значення, отримане функцією стиснення на попередньому кроці, у нове l -бітне значення.

Далі використовуються наступні позначення:

- $\oplus$ – додавання за модулем 2 (XOR);
- $0x$ – префікс числа, що записане у шістнадцятковій системі числення;
- $a \bmod b$ – ціле невід’ємне число, що дорівнює залишку від ділення цілого числа a на натуральне число b ;
- B_i – i -й байт вхідної послідовності;
- C^i – константа перетворення XORRoundKey або Add64RoundKey для i -го циклу;
- c – кількість стовпців внутрішнього стану в матричному поданні;
- ϕ – функція стиснення;
- H – визначена у стандарті функція гешування;
- $H(M)$ – результат обчислення функції гешування для повідомлення M (гешзначення);
- IV – вектор ініціалізації;
- l – розмір внутрішнього стану функції гешування (у бітах), $l \in \{512, 1024\}$;
- M – повідомлення;

					ВКРБ-123.26.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

- m_i – i -й блок повідомлення M ;
- n – довжина обчисленого геш-значення;
- N – довжина повідомлення M без доповнення;
- P, Q – складові перетворення функції стиснення;
- P_{512} – перетворення P для 512-бітного внутрішнього стану;
- P_{1024} – перетворення P для 1024-бітного внутрішнього стану;
- Q_{512} – перетворення Q для 512-бітного внутрішнього стану;
- Q_{1024} – перетворення Q для 1024-бітного внутрішнього стану;
- r – кількість ітерацій у перетвореннях P і Q ($r = t$ в ДСТУ 7564:2014);
- S – внутрішній стан геш-функції;
- t – кількість блоків m , з яких складається повідомлення M , включаючи доповнення;
- v_i – i -й біт вхідної послідовності;
- Ω – завершальне перетворення;
- Купина- n – режим використання функції гешування з усіченням обчисленого гешзначення до розміру n біт.

Загальні положення

Під функцією гешування H розуміється залежне від вектора ініціалізації IV відображення послідовності біт M у геш-значення $H(M)$ фіксованої довжини n .

ДСТУ 7564:2014 визначає функцію гешування, яка виконує перетворення «Купина-256» або «Купина-512», що забезпечують обчислення геш-значення з довжинами 256 або 512 біт відповідно.

Геш-значення довжиною 256 бітів додатково може бути усічено до бітової послідовності довжиною від 8 до 248 біт з кроком у 8 біт, 512 бітів може бути усічене до бітової послідовності довжиною від 264 до 504 біт з кроком у 8 біт.

Режим роботи для формування геш-значення довжиною n біт позначається як «Купина- n ».

Основними режимами роботи функції ґешування, що рекомендуються до застосування, є «Купина-256», «Купина-384» і «Купина-512».

Структура перетворення

Функція ґешування, визначена в ДСТУ 7564:2014, формує ґеш-значення для повідомлення, що складається з бітової послідовності довжини від 0 біт (порожній рядок) до $2^{96}-1$ біт.

При формуванні ґеш-значення повідомлення доповнюється, далі поділяється на l -бітні блоки $m_0, \dots, m_t$, після чого виконується обробка кожного блоку шляхом ітеративного виконання функції стиснення φ .

При цьому формуються значення $h_i = \varphi(h_{i-1}, m_i)$ де $i = 1, \dots, t$, а початкове значення $h_0 = IV$.

Після обробки останнього блоку повідомлення результуюче ґеш-значення обчислюється як $H(M) = \Omega(h_t)$, де Ω – завершальне перетворення, що повертає n - бітне значення, кратне 8 ($0 < n \leq l/2$).

					ВКРБ-123.26.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено розроблене у бакалаврської дипломної роботі система Legrand Cabling System 3 для ЦОД. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи: Функціональних кнопок ПЗ; Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші; Розділу обрання групи; Розділу виведення результату роботи системи.

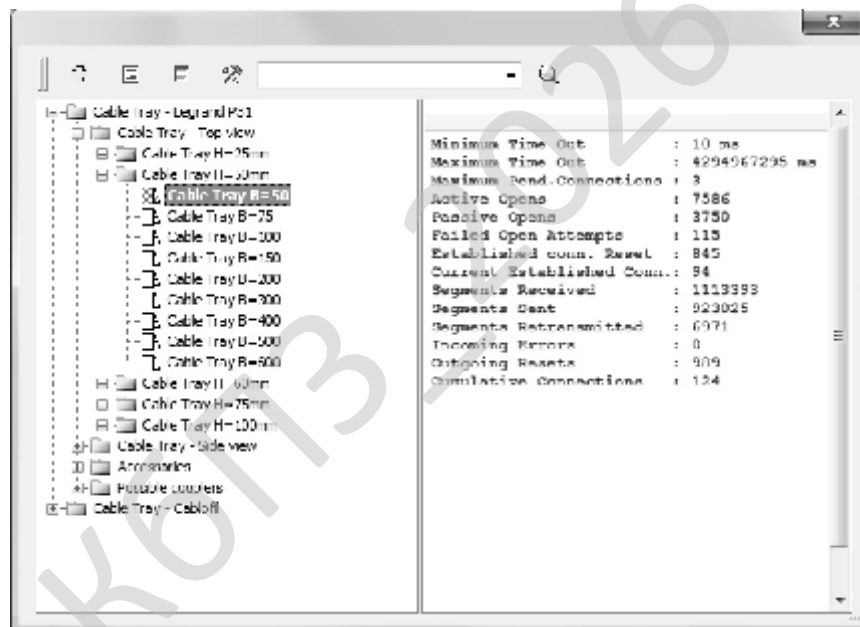


Рисунок 5.1 – Головне вікно ПЗ

Розроблена програма має дуже простий і 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

помилки, і працює, то можна використовувати, інакше слід слідувати інструкціям, які пропонує програма.

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

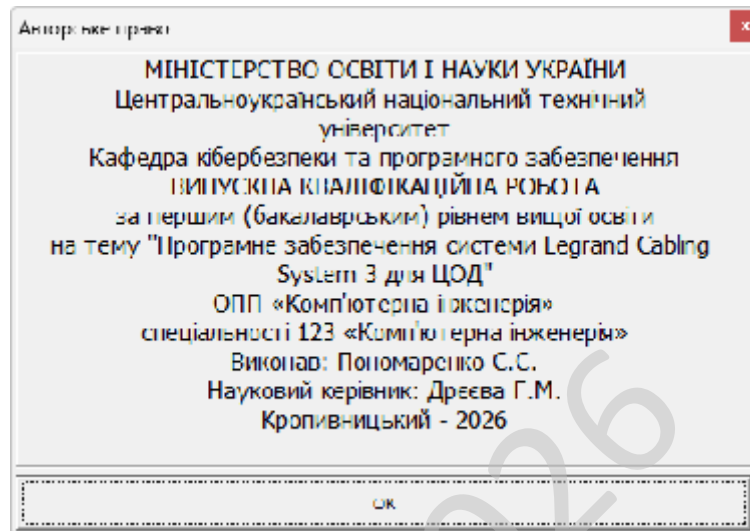


Рисунок 5.2 – Авторське право

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":

- 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.

Проводилось тестування чорної скриньки. Основне місце програми тестів «чорної скриньки» — інтерфейс ПЗ. Відомі: функції програми. Досліджується: робота кожної функції на всій області визначення.

Ці тести демонструють:

- Як виконуються функції програми.
- Як приймаються вихідні дані.
- Як виробляються результати.
- Як зберігається цілісність зовнішньої інформації.

При тестуванні «чорної скриньки» розглядаються системні характеристики програм, ігнорується їхня внутрішня логічна структура. Вичерпне тестування, як правило, неможливе.

Наприклад, якщо в програмі 10 вхідних величин і кожна приймає по 10 значень, то кількість тестових варіантів становитиме 10^{10} . Тестування «чорної скриньки» не реагує на багато особливостей програмних помилок.

Тестування «чорної скриньки» (функціональне тестування) дозволяє отримати комбінації вхідних даних, які забезпечують повну перевірку всіх функціональних вимог до програми.

Програмний виріб тут розглядається як «чорна скринька», чию поведінку можна визначити тільки дослідженням його входів та відповідних виходів. При такому підході бажано мати:

- Набір, утворений такими вхідними даними, які призводять до аномалій у поведінці програми (назвемо його ІТс).
- Набір, утворений такими вхідними даними, які демонструють дефекти програми (назвемо його ОТ).

Будь-який спосіб тестування «чорної скриньки» повинен:

- Виявити такі вхідні дані, які з високою ймовірністю належать набору ІТ.

					ВКРБ-123.26.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

– Сформулювати такі очікувані результати, які з високою імовірністю є елементами набору ОТ.

Принцип «чорної скриньки» не альтернативний принципу «білої скриньки». Скоріше це доповнює підхід, який виявляє інший клас помилок.

Тестування «чорної скриньки» забезпечує пошук наступних категорій помилок:

- Некоректних чи відсутніх функцій.
- Помилки інтерфейсу.
- Помилки у зовнішніх структурах даних або в доступі до зовнішньої бази даних.
- Помилки характеристик (необхідна ємність пам'яті і т.д.).
- Помилки ініціалізації та завершення.

Обрано умови розповсюдження – commercial software. Програмне забезпечення, створене комерційною організацією з метою отримання прибутку від його використання іншими, наприклад, шляхом продажу копій.

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

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

Далеко не завжди, але як правило терміни критично важливих змін в комерційних продуктах значно менше, ніж у некомерційних проєктів. Це пов'язано з тим, що над комерційним продуктом працюють цілі групи розробників і ця робота є їх основним заняттям.

Розробникам-початківцям як правило доводиться шукати додаткові способи заробітку, і це збільшує час, що витрачається на доповнення і зміни програм.

					ВКРБ-123.26.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

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

Окремий вид комерційних програм, коли їх розробка оплачується безпосередньо замовником.

Такі програми найчастіше позбавлені всіх переваг комерційних продуктів, оскільки мають обмежений бюджет, але більш адаптовані до вимог замовника, ніж аналоги.

К6ПЗ_2026

					ВКРБ-123.26.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи Legrand Cabling System 3 для ЦОД.

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

Рішення завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем Legrand Cabling System 3 для ЦОД.
- Досліджена система Legrand Cabling System 3 для ЦОД.
- На основі отриманих результатів досліджень створена програмна реалізація системи Legrand Cabling System 3 для ЦОД.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання Legrand Cabling System 3 для ЦОД.

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 призначені для системи Legrand Cabling System 3 для ЦОД. 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

					ВКРБ-123.26.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

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 ДСТУ 7564:2014.

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.

КБПЗ-2026

					ВКРБ-123.26.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Scott Jernigan «CompTIA Network+ Certification All-in-One Exam Guide, Eighth Edition». 2022. – 976 p.
2. Doug Lowe «Networking For Dummies 12th Edition». 2020. – 480 p.
3. Ramon Nastase «Computer Networking: The Beginner's guide for Mastering Computer Networking, the Internet and the OSI Model». 2018. – 186 p.
4. Russ White & Ethan Banks «Computer Networking Problems and Solutions: An Innovative Approach to Building Resilient, Modern Networks». 2017. – 832 p.
5. Вінтенко Б., Смірнов О., Миронець І., Смірнова Т., Смірнов С. «Імітаційна модель шляхів вхідних даних комп'ютерної інтелектуальної системи підтримки оператора енергоблоку АЕС». *Комбінаторні конфігурації та їхні застосування: Матеріали XXVII Міжнародного науково-практичного семінару, присвяченого 125-річчю Національного університету «Запорізька політехніка»* (Запоріжжя-Кропивницький-Київ, 4-6 червня 2025 р.). Запоріжжя: НУ «Запорізька політехніка», 2025. С.82-91.
6. Al-Azzeh, J., Ayyoub, B., Mesleh, A., Smirnova, T., Gnatyuk, S., Drieiev, O., Smirnov, O., Dorenskyi, O. «Cloud-Based Information System for Evaluating Caverns in the Process of Blasting Metal Surfaces of Details». *International Review on Modelling and Simulations* 18 (1), 2025. pp. 32-42.
7. Смірнова Т.В., Коноплицька-Слободенюк О.К., Буравченко К.О., Смірнов С.А., Кравчук О.В., Козірова Н.Л., Смірнов О.А. «Дослідження технологій забезпечення кібербезпеки хмарних сервісів IaaS, PaaS та SaaS». *Кібербезпека: освіта, наука, техніка*. 2024. №4(24), С. 6-27.
8. Батрак О., Смірнова Т., Гнатюк В., Одарченко Р., Смірнов О. «Дослідження показників ефективності функціонування та перспектив розвитку систем ІР-телефонії». *Підводні технології*, 2024, № 13, с. 28-35.

9. Kuznetsov, O., Kryvinska, N., Ilchenko, O., Smirnova, T., Ulianovska, Y. «Comparative Analysis of Cryptocurrency Trading Platforms Using the Analytic Hierarchy Process». *CEUR Workshop Proceedings*, 2023, 3628, pp. 106-115.

10. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56.

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

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

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

14. Smirnova, T., Gnatyuk, S., Yudin, O., Sydorenko, V., Polozhentsev, A., «The Model for Calculating the Quantitative Criteria for Assessing the Security Level of Information and Telecommunication Systems». *CEUR Workshop Proceedings* Volume 3156, 2022, Pages 390-399.

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

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

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

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

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

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

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

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

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

23. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 48. Springer, Cham. 2021. pp 557-587.
24. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». *CEUR Workshop Proceedings* Volume 2616, 2020, Pages 125-136.
25. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». *CEUR Workshop Proceedings* Volume 2616, 2020, Pages 366-379.
26. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», *CEUR Workshop Proceedings* Volume 2608, 2020, Pages 633-645.
27. Smirnov O. Kuznetsov A., Zaichenko Yu., Pastukhov M., Oleshko O., Kuznetsova K., «Formation of Discrete Signals with Special Correlation Properties». *International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2019*; Odessa; Ukraine; 9-13 September 2019. P.22-28.
28. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». *International Journal of Computing*; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.
29. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*, Kyiv, Ukraine, 8 – 11 October 2019 . P.517-522.

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

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

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

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

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

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

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

37. Смірнов О.А., Смірнова Т.В., Буравченко К.О., Кравченко С.С., Горбов В.О., «Хмарна система підтримки прийняття рішень технологічного

процесу відновлення поверхонь конструкцій і деталей машин». *Сучасні інформаційні системи*. 2021. Т. 5, № 4. С. 79-95

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

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

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

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

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

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

44. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan 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).

					ВКРБ-123.26.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

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

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

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

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

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

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

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

52. Смірнов О.А., Смірнов С.А. Дідик А.К., Дреєв О.М. Моделі системи нейромережових експертів безпечної маршрутизації у хмарних

					ВКРБ-123.26.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

антивірусних системах. Збірник наукових праць "Системи обробки інформації". - Випуск 3 (140). - Х.: ХУПС - 2016. - С. 36-39.

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

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

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

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

					ВКРБ-123.26.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69