

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

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

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи керування
мережною інфраструктурою на основі Cisco Application Centric
Infrastructure”

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

Керівник проекту
кандидат технічних наук, доцент
_____ Дреєв О.М.
« ____ » _____ 2024 р.
Рецензент _____

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

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 6 » вересня 2024 року

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

Якименку Сергію Олександровичу

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та програмна реалізація системи керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure

2. Керівник роботи Дреєв Олександр Миколайович, канд. техн. наук, доцент

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

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

3. Строк подання студентом роботи до захисту 2.12.2024 р.

4. Мета та завдання випускної кваліфікаційної роботи: Метою розробки є дослідження та програмна реалізація системи керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure

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

1. Призначення та область використання.

6. Наукова новизна.

2. Перегляд аналогічних існуючих систем.

7. Маркетингове та економічне обґрунтування IT-проєкту.

3. Опис і обґрунтування проектних рішень.

8. Заходи з охорони праці та техніки безпеки.

4. Етапи програмування системи.

9. Висновки.

5. Впровадження системи в промислову експлуатацію

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

Наукова новизна 1 аркуш

Структурна схема системи 1 аркуш

Функціональна схема системи 1 аркуш

Діаграма процесів 1 аркуш

Блок-схема алгоритму роботи додатку 2 аркуша

Показники економічної ефективності 1 аркуш

6. Консультанти розділів роботи

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

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

КАЛЕНДАРНИЙ ПЛАН

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

Дата видачі завдання
« 6 » вересня 2024 р.

Підпис керівника

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

Підпис здобувача

(прізвище та ініціали)

АНОТАЦІЯ

Якименко С.О. Дослідження та програмна реалізація системи керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure. 123 Комп'ютерна інженерія. Центральнoукраїнський національний технічний університет. Кропивницький. 2024.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure.

Метою розробки є дослідження та програмна реалізація системи керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure.

Об'єктом дослідження є процес керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure.

Предметом дослідження є методи керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure.

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

Результат роботи – програмна реалізація системи керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure.

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 Builder C++.

Ключові слова: комп'ютерна інженерія, Cisco Application Centric Infrastructure

ABSTRACT

Yakymenko S.O. Research and software implementation of a network infrastructure management system based on Cisco Application Centric Infrastructure. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2024.

In this final qualification work for the second (master's) level of higher education, software has been developed, which is intended for a network infrastructure management system based on Cisco Application Centric Infrastructure.

The purpose of the development is the research and software implementation of a network infrastructure management system based on Cisco Application Centric Infrastructure.

The object of the research is the process of managing network infrastructure based on Cisco Application Centric Infrastructure.

The subject of the research is methods of managing network infrastructure based on Cisco Application Centric Infrastructure.

The research methods are based on methods of computer network theory, methods of mathematical statistics, methods of software development.

The result of the work is a software implementation of a network infrastructure management system based on Cisco Application Centric Infrastructure.

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 user-friendly user interface has been developed. Instructions for working with the software are provided.

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

The program was developed in the Builder C++ environment.

Keywords: computer engineering, Cisco Application Centric Infrastructure

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	7
1.1 Призначення системи.....	7
1.2 Область застосування.....	8
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	9
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	9
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	14
2.3 Розгорнута постановка завдання	16
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	18
3.1 Опис функціонування системи	18
3.2 Розробка структурної схеми.....	24
3.3 Розробка функціональної схеми	29
3.4 Розробка діаграми процесів.....	39
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	41
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	41
4.2 Захист розробленого програмного забезпечення.....	57
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	60
6 НАУКОВА НОВИЗНА	67

					ВКРМ-123.24.0050.00.00.ПЗ			
Вим.	Арк.	№ докум.	Підп.	Дата	Дослідження та програмна реалізація системи керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure	Літ.	Аркуш	Аркушів
Розроб.	Якименко С.О.					М	1	93
Перев.	Дресв О.М.					ЦНТУ КІ-23М		
Н.контр.	Коваленко А.С.							
Затв.	Смірнов О.А.							

7	МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ІТ-ПРОЄКТУ	68
7.1	Визначення цільової аудиторії кінцевого готового продукту	68
7.2	Оцінка привабливості шляхом застосування методів експертних оцінок ...	69
7.3	Вибір методу оцінки вартості ПЗ	70
7.4	Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості.....	71
7.5	Пропозиція алгоритму просування проєкту розробки ПЗ	71
7.6	Оптимізація каналів збуту та шляхів реалізації ПЗ	73
7.7	Визначення ключових факторів успіху конкретного проєкту.....	75
8	ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	76
8.1	Вступ.....	76
8.2	Аналіз умов праці на робочому місці фахівця	78
8.3	Розробка заходів з умов поліпшення охорони праці.....	81
8.4	Розрахункова частина	82
8.5	Висновки до розділу.....	84
9	ОСНОВНІ ВИСНОВКИ.....	85
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	87

КБПЗ-2024

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

АСУ	–	автоматизована система управління
АЦП	–	аналогово-цифровий перетворювач
КПК	–	кишеньковий персональний комп'ютер
ПЗ	–	програмне забезпечення
ПЗО	–	пристрій зв'язку з об'єктом
ПК	–	персональний комп'ютер
СЗД	–	системи зберігання даних
СУБД	–	системи управління базами даних
ТП	–	технологічний процес
НМІ	–	людино-машинний інтерфейс
SCADA	–	диспетчерське управління й збір даних
WAP	–	Wireless Application Protocol

КБПЗ – 2024

ВСТУП

Актуальність теми. У концепції Cisco Application Centric Infrastructure (ACI) пропонується принципово новий підхід до побудови мережної інфраструктури центрів обробки даних. Чим же ця архітектура відрізняється від інших, як традиційних, так і недавно з'явилися? І як інтегрувати в неї елементи інших виробників і рішення Open Source? Архітектура ACI складається із трьох основних компонентів: моделі політик, мережної фабрики й контролера APIC, за допомогою якого прийнята політика доводить до елементів фабрики й здійснюється контроль за її функціонуванням. Нижче ми розглянемо, які ключові особливості кожного із цих компонентів і як з них формується єдине рішення. Назва Application Centric Infrastructure можна перевести як «інфраструктура, орієнтована на додатки». Тим самим підкреслюється, що в главу кута поставлена не мережна, а прикладна ієрархія центра обробки даних – будь те ЦОД традиційного корпоративного замовника або оператора хмарних послуг. Безумовно, у будь-якому мережному рішенні для ЦОД так чи інакше враховуються вимоги додатків, але, як правило, це досягається шляхом їхнього співвіднесення з мережною номенклатурою – адресами, підмережами, VLAN і т.д. У результаті прийнята політика (безпеки, якості обслуговування й т.д.) звичайно застосовується на границі між підмережами, причому підмережі й конкретні адреси в них еквівалентні класифікаторам серверів у рамках даної політики, а політика, у свою чергу, виражається у вигляді списків доступу (ACL) з обліком тих або інших мережних атрибутів. Таке положення справ приводить до того, що при впровадженні додатків всі вимоги доводиться «переводити» з мови системних адміністраторів на мову мережних адміністраторів, а сам процес виявляється негнучким, повільним і підданим помилкам.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure.

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

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

– Огляд існуючих систем керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure.

– Дослідження системи керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure.

– Програмна реалізація системи керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure.

Об'єктом дослідження є процес керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure.

Предметом дослідження є методи керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure.

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

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

– Удосконалено метод керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure.

– Розроблено вітчизняний продукт керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure.

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

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

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

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

КБПЗ_2024

					VKPM-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Прикладом моделі політик може служити конфігурація міжмережного екрана, типова для більшості великих організацій: вона містить тисячі або навіть десятку тисяч рядків, і найчастіше про призначення багатьох з них уже ніхто не пам'ятає. На противагу цьому АСІ описує політику для додатків у термінах самих додатків і їхніх компонентів. Як ключові елементи для опису політики в інфраструктурі АСІ використовуються мережні профілі додатків (Application Network Profile), у яких задаються правила взаємодії між додатками або рівнями додатка. Останні представляються у вигляді груп елементів (End-Point Group, EPG), що входять до складу додатка.

Політика, представлена в рамках мережного профілю додатка, описується в максимально абстрактних термінах – для цього не залучаються ні використовувані адреси, ні кількість елементів, що входять у групу, ні їхній тип (фізичні сервери, віртуальні машини або їхня комбінація). З одного боку, така політика може бути сформульована безпосередньо розроблювачами й адміністраторами додатків. Даний підхід відповідає широко обговорюваній в останні роки моделі DevOps, відповідно до якої адміністратори додатків повинні взаємодіяти з інфраструктурою прямо, а не за допомогою серверних і мережних адміністраторів. З іншого боку, це дозволяє оперувати профілями додатків як логічними, абстрактними об'єктами, що допускають, скажемо, не тільки тиражування додатка різними замовниками при впровадженні декількох однотипних копій, але й копіювання між декількома ЦОД, незважаючи на використання різної адресації, розбіжність числа серверів і т.д.

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

1.2 Область застосування

Політика взаємодії між групами ERG може передбачати не просто заборону або дозвіл трафіку між вхідними в них елементами, але й більше складні операції з боку фабрики – зокрема, формування копій трафіку або перенапрямок трафіку на послідовність сервісних елементів, що виконують, наприклад, функції балансування навантаження й міжмережного екранування. Пошук ефективних механізмів для вбудовування таких сервісів у мережну інфраструктуру ЦОД довгий час залишався одним із ключових завдань її проектування. У більшості інсталяцій для цього використовується «зшивання VLAN» за допомогою сервісних елементів, однак такий підхід у крайній негнучкості й погано масштабуємо.

В АСІ сервісні елементи включаються в шлях трафіку автоматично, засобами самої мережної інфраструктури, причому це можуть бути елементи розробки як Cisco, так і сторонніх компаній, як віртуальні, так і реалізовані у вигляді окремих пристроїв. Більше того, інструментарій для опису сервісних елементів у форматі, «зрозумілому» АСІ, досить простий і може бути використаний сторонніми розроблювачами й навіть ІТ-фахівцями компанії замовника.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

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

З ростом числа обчислювальних компонентів, що включаються до складу сучасних високопродуктивних обчислювальних систем (ВОС), що існують підходи, методи й засоби для організації системного ПЗ ВОС перестають задовольняти вимогам. Аналіз існуючих технологій створення й застосування високопродуктивних ВОС петафлопного рівня (див., наприклад, роботи [1, 5]), а також сучасних тенденцій до побудови ВОС ексафлопного рівня (див., наприклад, роботи [3, 21, 32]) дозволяє зробити висновок про те, що надійність таких систем у рамках використовуваних на сьогоднішній день парадигм і технологій буде досить низкою. У зв'язку із цим актуальним і перспективним напрямком досліджень в області системного ПЗ ВОС є створення нових підходів, методів і засобів керування й моніторингу ВОС, здатних забезпечувати необхідний рівень відкозостійкості й надійності для обчислювальних середовищ такого масштабу.

У завдання засобів керування ВОС входить розподіл навантаження на обчислювальні вузли, виконання непаралельних і паралельних команд із наступною передачею результату користувачеві або операторові, виконання завантаження й зупинки роботи обчислювальних вузлів і ряд інших. У завдання засобів моніторингу входить збір даних про роботу всіх компонентів ВОС, багатокритеріальний аналіз даних, що збираються, і у випадку виявлення яких-небудь відхилень прийняття необхідних впливів на компоненти.

Моніторинг компонентів ВОС умовно можна розділити на наступні категорії:

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

– моніторинг і аналіз ефективності виконання програм у ВОС (контроль поточного стану обчислювальних процесів і їхніх окремих екземплярів, оцінка ефективності використання виділених ресурсів);

– моніторинг, тестування й діагностика апаратних компонентів обчислювальних вузлів (жорсткі диски, процесори, оперативна пам'ять, мережні інтерфейси й ін.);

– моніторинг інженерної інфраструктури ВОС (системи безперебійного живлення, кліматичне встаткування, системи пожежогасіння й ін.);

– моніторинг обчислювальної інфраструктури ВОС (моніторинг поточного завантаження обчислювальних вузлів, контроль комунікаційних, керуючих і сервісних мереж, систем зберігання даних);

– моніторинг проміжного програмного забезпечення ВОС (моніторинг функціонування системних служб, черг завдань, агентів, різних підсистем і ін.).

Розробка комплексної системи моніторингу, що забезпечувала б збір даних з величезної кількості різнорідних компонентів, що входять до складу сучасних ВОС, є тяжкорезалізуємым завданням через відсутність стандартизованих форматів і протоколів збору даних із усього безлічі різнорідних програмно-апаратних компонентів ВОС. З іншого боку, на сьогоднішній день існує величезна кількість програмних рішень, які окремо дозволяють забезпечувати моніторинг необхідних компонентів ВОС. Більше того, багато компонентів ВОС уже постачені системами локального моніторингу. У зв'язку із чим найбільш доцільним і перспективним напрямком розвитку досліджень по створенню комплексних систем моніторингу ВОС є агрегація існуючих локальних систем моніторингу в рамках комплексної системи позначка-моніторингу ВОС [9]. При цьому локальна система моніторингу виступає лише постачальником даних, а їхній експертний аналіз і прийняття на основі результатів аналізу необхідних регулюючих впливів приділяється системі позначки-моніторингу.

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

– HPCToolkit [16] – інструментарій для моніторингу виконання паралельних програм з можливістю аналізу використовуваних ресурсів, автоматичного виявлення неефективних блоків із прив'язкою до вихідного тексту програми.

– Paraver [28] – аналізатор продуктивності програм, заснований на трасуванні подій і що дозволяє робити детальний аналіз зміни й розподілу метрик з метою розуміння поведження додатків.

– Scalasca [30] – інструментарій для виконання оптимізації паралельних програм шляхом виміру й аналізу їхнього поведження під час виконання.

Із представленого списку найбільш функціональними й перспективними рішеннями для аналізу ефективності виконання паралельних програм у ВОС, з погляду авторів, є пакети з відкритим вихідним кодом NWPerf і Paraver.

Моніторинг, тестування й діагностика апаратних компонентів обчислювальних вузлів

Для виявлення несправностей в апаратних компонентах обчислювальних вузлів ВОС, на жаль, авторам відомо лише невелике число засобів. З них заслуговують на увагу:

– Disparity [11] – підхід, заснований на запуску MPI-програми на аналізованих вузлах з метою виявлення можливих несправностей.

– CIFTS [13] – використовує механізми обміну інформацією про несправності з метою виробітку цілісної картини про стан вузлів.

Найцікавішим з них є програмний засіб Disparty, що дозволяє виявляти несправності компонентів обчислювального вузла під час простою між запусками екземплярів обчислювальних процесів. У даній категорії авторами розробляється власний підхід [31] для багатоступінчастої діагностики обчислювальних вузлів з використанням набору стандартних утиліт (Sensors, SMART, IMPUtils і ін.).

Моніторинг інженерної інфраструктури ВОС

Для моніторингу інженерної інфраструктури суперкомп'ютерних центрів і центрів обробки даних найпоширенішими засобами є: Clustr [10], EMC ViRP

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

SRM [12], HP Cluster Management Utility [15], Bright Cluster Manager [8], Moab Adaptive HPC Suite [23], IBM cluster system management [17]. Однак всі перераховані засоби є пропрієтарними, найчастіше жорстко прив'язані до встаткування й не завжди мають достатню гнучкість для моніторингу інфраструктури різнорідних обчислювальних середовищ.

Серед некомерційних продуктів авторам не вдалося знайти прийнятних засобів, які дозволяли б універсально описувати состав різнорідного інженерного встаткування суперкомп'ютерного центра, створювати нові об'єкти й задавати правила їхнього моніторингу. Системи моніторингу Nagios [20], Zabbix [35] надають набір засобів для моніторингу інфраструктури ВОС, які в кожному окремому випадку необхідно істотно допрацьовувати. У даній категорії авторами також здійснюється розробка власних універсальних засобів, що базуються на агрегації локальних систем моніторингу інженерного встаткування. Деякі аспекти реалізації розроблювального авторами засобів моніторингу інженерної інфраструктури наведені в роботі [2].

Моніторинг обчислювальної інфраструктури ВОС

У даній категорії на сьогоднішній день існує значне число комплексних рішень. Найбільш популярними з них є:

– Ganglia [22] – масштабована розподілена система моніторингу кластерів паралельних і розподілених обчислень і хмарних систем з ієрархічною структурою;

– Nagios – система моніторингу обчислювальних систем і мереж із широкими можливостями повідомлення оператора про можливі несправності;

– Zabbix – система моніторингу й відстеження стану різноманітних сервісів комп'ютерної мережі, серверів і мережного встаткування;

– ZenOSS [36] – система моніторингу з можливостями автоматичного виявлення й конфігурування параметрів контролю різних систем;

– Ovis2 [27] – комплексна система моніторингу, що забезпечує високу масштабованість і інтеграцію з іншими системами моніторингу.

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

Найбільш популярною системою в даній категорії як і раніше є система моніторингу Ganglia. Однак стандартний набір функцій даної системи не забезпечує зростаючих потреб по моніторингу обчислювального встаткування ВОО, що найчастіше приводить до необхідності використання додаткових систем моніторингу, таких як Zabbix або Nagios. Найбільш перспективною системою в даній категорії, з погляду авторів, є система Ovis2, що забезпечує високу масштабованість і широкі можливості по підключенню різних джерел даних.

Моніторинг проміжного програмного забезпечення ВОО

У даній категорії можуть використовуватися як описані вище системи моніторингу Nagios і Zabbix, так і більше спеціалізовані засоби, такі як:

- Хумон [34] – моніторинг роботи системних сервісів;
- FATE [14] – інструментарій тестування хмарних додатків;
- CloudRift [29] – середовище тестування мікросервісних додатків.

Крім перерахованих засобів, адміністраторами ВОО найчастіше розробляються спеціалізовані утиліти для відстеження коректного функціонування окремих підсистем, що входять до складу проміжного ПЗ ВОО, найчастіше реалізовані у вигляді скриптів, що запускаються за розкладом з використанням сервісу CRON.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Оскільки потрібно розробити просту та легку у користуванні програму, яка б виконувалась під операційною системою Windows, то для її реалізації я обрав Builder C++. Існує велике число бібліотек написаних під Builder C++ , тому це одна з важливих причин вибору мови програмування. Середовище Builder C++ досить просте в користуванні, його вихідний код значно менше по об'єму в порівнянні з Delphi чи деякими іншими програмами такого типу. Досить легко

						ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			14

організувати взаємодію між модулями програм, об'єктно-орієнтований підхід дає можливість значно скоротити код програми, а отже і час його виконання.

На заміну старого розробленого набору елементів управління у Builder C++ інтегрована бібліотека візуальних компонентів VCL, представлених на палітрі компонентів. Після переносу на форму методом перетягування (drag-and-drop) компоненти відразу становляться діючими об'єктами вашої програми. Окрім типізованих інтерфейсних елементів Windows (кнопки, смуги прокручування, редагуємі текстові області, прості та комбіновані списки, та інше) у бібліотеку включені елементи підтримки діалогових вікон, обслуговування баз даних та багато іншого. Можливо не тільки модифікувати поведінку існуючих компонентів, але і будувати нові.

Builder C++ підтримує останні розширення стандарту мови C++ та забезпечує швидку компіляцію та складання 32-розрядних програм для Windows. Результуючі програми оптимізовані з точки зору швидкості виконання програм та затрат пам'яті. Зручний відлагоджувальник (з асемблерним вікном, можливістю крокового виконання, завдання точок зупинки, трасування та інше) повністю інтегрований у систему проектування. Дизайнер форм, редактор коду, інспектор об'єктів та інші інструменти зостаються доступними під час виконання програми, саме через це вносити зміни до коду можна прямо у процесі відлагодження.

Дизайнер форм, Інспектор об'єктів і інші засоби залишаються доступними під час роботи програми, тому вносити зміни можна в процесі відлагодження.

Builder C++ поставляється в трьох варіантах: Standard (стандартний), Professional (для професіоналів розробників, орієнтованих на мережеву архітектуру) і Client/Server Suite (для розробки систем в архітектурі клієнт/сервер). Останні два варіанти доповнюють стандартний початковими текстами візуальних компонентів, різномасштабним словником даних, новими функціями мови запитів SQL для бази даних, пакетом підтримки систем Internet, службою моніторингу програм, а також рядом інших засобів.

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

Builder C++ підтримує зв'язок з різними базами даних 3-х видів: dBASE і Paradox; Sybase, Oracle, InterBase і Informix; Excel, Access, FoxPro і Vtrieve. Механізм BDE (Borland Database Engine) додає обслуговуванню зв'язків з базами даних дивовижну простоту і прозорість. Провідник Database Explorer дозволяє зображати зв'язки і об'єкти баз даних графічно. Використовуючи компоненти баз даних, я побудував електронний записник згідно таблиці dBASE за півгодини роботи на комп'ютері. Спадкоємство готових форм і їх "підгонка" під специфічні вимоги помітно скорочують тимчасові витрати на вирішення подібних завдань.

Довідкова служба Builder C++ надавала мені допомогу в цій і багатьох інших подібних ситуаціях. Є повний опис кожного управляемого компонента, включаючи списки властивостей і методів, а також численні приклади. Виклад матеріалу в книзі був значно покращуваний і систематизований завдяки відомостям, почерпнутим мною з довідкової служби.

Завдяки засобам управління проектами, двосторонній інтеграції додатку і синхронізації між засобами візуального і текстового редагування, а також вбудованому відладнику (з асемблерним вікном прокрутки, покрокового виконання, точок останову, трасуванням і тому подібне) – Builder C++ корпорації Borland надає собою вражаюче середовище розробки, яка, мабуть, витримає конкурентну боротьбу з такими модними продуктами як Developer Studio фірми Microsoft.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure.

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

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

- а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;
- б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;
- в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;
- г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;
- д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;
- е) провести розрахунки по визначенню економічної ефективності розробленої системи;
- ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;
- з) сформулювати висновки про виконаний обсяг робіт та одержані результати.

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Мережна фабрика

З погляду властиво мережного рішення ACI являє собою комутуючу матрицю, побудовану відповідно до дворівневої мережної топології Клоза, що складає з «листів» (leaf) – пристроїв доступу, до яких підключаються сервери, сервісні елементи й канали, що з'єднують із зовнішнім миром, – і «хребта» (spine), завданням якого є комутація між «листами», а також виконання службових функцій по реалізації накладеної мережі.

На даний момент для побудови фабрики ACI необхідно використовувати комутатори сімейства Cisco Nexus 9300/9500, спеціально розроблені для реалізації функцій ACI. Вони забезпечують, зокрема, застосування політики на основі EPG, створення накладеної мережі, нормалізацію інкапсуляції, динамічне балансування трафіку у фабриці, збір детальної телеметричної інформації й багато чого іншого. Трафік усередині фабрики передається по високопродуктивних з'єднаннях 40 Gigabit Ethernet, причому для ACI були розроблені двонаправлені трансівери 40Gb, що дозволяють передавати 40Gb по звичайній парі багатомодових волокон, тобто перейти з 10G на 40G без внесення яких-небудь змін у кабельну систему (ця розробка доступна тепер і в інших продуктах Cisco). Для підключення серверів фабрика ACI надає інтерфейси 1G, 10G і 40G, причому система побудована так, щоб перехід зі звичних з'єднань 1G на 10G відбувався без помітного збільшення ціни.

Особливої уваги заслуговує логіка передачі даних усередині фабрики ACI. Історично адресація в протоколі IP застосовувалася для виконання відразу двох різних завдань – ідентифікації (вказівки на конкретний вузол, що є відправником або одержувачем пакета) і локалізації (вказівки на місце розташування за

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

АСІ використовує інтегровану в мережну фабрику технологію накладених мереж, що дозволяє будувати динамічно формовані сегменти з комутацією другого й третього рівня поверх маршрутизуємої транспортної мережі. При цьому сервери можуть використовувати всілякі типи підключень – звичайні VLAN, інкапсуляцію VXLAN або NVGRE. На границі фабрики виконується «нормалізація» інкапсуляції в єдиний для фабрики формат eVXLAN, побудований на базі VXLAN з передачею в заголовках додаткової інформації. У результаті з'являється можливість об'єднати в одному логічному сегменті віртуальні машини, розташовані на хості під керуванням VMware vSphere, де застосовується інкапсуляція VXLAN, і на хості під керуванням Microsoft Hyper-V з інкапсуляцією NVGRE.

До інших переваг технології накладеної мережі, використовуваної в АСІ, ставляться висока продуктивність інкапсуляції, що досягається за рахунок апаратної реалізації, а також надзвичайно висока масштабованість і продуктивність процесу відображення адрес накладеної мережі в транспортні адреси.

Контролер АРІС

Централізоване керування інфраструктурою АСІ здійснює контролер Application Policy Infrastructure Controller (APIC). У дійсності він являє собою кластер контролерів, оскільки для цілей масштабування й забезпечення безперервності роботи АРІС підтримує кластеризацію. Незважаючи на певну аналогію з контролером програмно обумовлених мереж (SDN), АРІС реалізує зовсім іншу модель керування.

У традиційній моделі SDN на центральному контролері, що реалізує функції рівня керуючої логіки (control plane) для всієї мережі, консолідується вся інформація про стан всіх мережних елементів, а його завданням є повне детальне налаштування всіх таблиць комутації на всіх мережних пристроях. Таким чином, можна говорити про те, що контролер SDN реалізує імперативну модель керування, у якій центральне керування всіма елементами інфраструктури

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

виконується шляхом точної й повної вказівки необхідних деталей налаштування. Такий підхід являє собою аналог «мікромеджменту» як стилю організаційного керування й приводить до появи пов'язаних із цим проблем масштабованості й у край високих вимог до надійності контролера.

В архітектурі АСІ функції адміністрування (management plane), тобто налаштування описаної вище мережної політики додатків, а також функцій моніторингу й діагностики, виконуються централізовано. При взаємодії з елементами мережі роль АРІС полягає в тому, щоб довести до них правила, а не деталі налаштування. У цьому змісті можна говорити про те, що АСІ використовує декларативну модель керування, коли мережі і її компонентів повідомляється, яке їхній стан є бажаним, а вони самостійно, без залучення ресурсів контролера, досягають його й підтримують у випадку яких-небудь змін у серверах і з'єднаннях, при міграції віртуальних машин і т.д.

Поза тим що АРІС реалізує централізоване керування через графічний інтерфейс адміністратора, він також забезпечує доступ через програмний інтерфейс (API) для вищестоящих систем керування, хмарної оркестрації, моніторингу й т.д. Останній, тобто «північний» інтерфейс між вищестоящими системами й контролером, повністю відкритий для використання сторонніми розроблювачами й замовниками, але цим «відкритість» АСІ не обмежується.

У галузі існує насущна потреба у використанні загальної мови декларативного опису політик інфраструктури. Разом з іншими компаніями Cisco опублікувала в IETF протокол OpFlex, де описується «південний» інтерфейс між контролером як крапкою опису правил і елементами, що реалізують ці правила, інфраструктури. Крім того, Cisco планує опублікувати референсну реалізацію з відкритим вихідним кодом агента OpFlex для віртуального комутатора, що дозволить полегшити й прискорити впровадження даного підходу в складі систем сторонніх розроблювачів.

У майбутньому реалізація OpFlex на стороні контролера планується не тільки в складі АСІ, тобто під управлінням контролера АРІС, але й у рамках

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

підтримку проекту має, що широку галузеву, контролера з відкритим вихідним кодом OpenDaylight, активним учасником якого є компанія Cisco. Таким чином, у перспективі можна буде говорити про можливість використання OpFlex як єдиного інструмента для поширення політики в інформаційних інфраструктурах.

Крім доведення політики до елементів фабрики, контролер APIC реалізує й потужні можливості по одержанню з фабрики діагностичної й телеметричної інформації. Причому й у цьому випадку використовується підхід, орієнтований на додатки, – адміністратор може побачити детальну інформацію не тільки про затримки, втрати й т.д. по кожному мережному елементі, але й про «стан здоров'я» інфраструктури з погляду конкретного додатка, що дозволяє спростити діагностику й усунення проблем, а також планування й оптимізацію продуктивності.

На закінчення відзначимо, що окремі елементи архітектури ACI доступні вже зараз, а вихід повного рішення очікується в середині 2017 року. У своєму нинішньому виді це рішення призначене для мережі ЦОД, але очевидна й потреба в розширенні моделі керування ACI як за межі ЦОД (у магістральну й кампусну мережа), так і в інші технологічні домени (обчислення й зберігання). Будемо сподіватися, що ця концепція буде розвиватися й далі.

Мікросегментація

Прикладом чутливості Cisco до вимог своїх клієнтів може служити розширення підтримки мікросегментації на віртуальні комутатори VMware (vDS або vSphere Distributed Switch) і Microsoft Hyper-V, а також на додатки, що працюють на встаткуванні bare-metal.

Мікросегментація дозволяє встановлювати гранулярні політики мережної безпеки на рівні L2 шляхом програмного «дроблення» мережі на велику кількість сегментів. Це дає можливість обмежити горизонтальне поширення погроз (шляхом реалізації так званих east-west Firewall) і поміщати в карантин скомпрометовані кінцеві крапки. Завдяки цьому досягається ізоляція злоумисника, що скомпрометували один вузол, від інших вузлів у тім же

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

VLAN. В АСІ призначення політик можливо по декількох атрибутах віртуальної машини – ім'я, гостьова ОС, ідентифікатор VM, FQDN, IP-адреса й др.

До сьогоднішнього дня Cisco пропонувала мікросегментацію тільки на базі Cisco AVS (Application Virtual Switch). Тепер функціонал розширений і на інші віртуальні й фізичні пристрої. Мікросегментація для Hyper-V доступна вже зараз, у той час як інші розширення з'являться в першому кварталі 2016 року.

Docker-контейнери

Cisco підтримує як віртуальні, так і апаратні прикінцеві крапки (Endpoints). У новому релізі з'явиться підтримка прикінцевих крапок, що працюють у контейнерах Docker. Функціонал реалізується за допомогою інтеграції контролера Cisco APIC (Application Policy Infrastructure Controller), що є компонентом АСІ, і Project Contiv – open source проекту, запущеного Cisco зовсім недавно. Метою Contiv є опис політик впровадження контейнерних додатків. Ці політики охоплюють область безпеки, інфраструктурні вимоги (дискова підсистема, виділення IP-адрес, ліміт смуги пропускання, вимоги до продуктивності, SLA), аналітикові й ін. Завдяки Project Contiv, тепер вони можуть бути поширені на додатки, що працюють в Linux-контейнерах.

Contiv включає мережний плагін до Docker, що дозволяє робити конфігурацію й підключення контейнера до фабрики АСІ. Також реалізований рівень прямої взаємодії з АРІ контролера АРІС.

Зараз ми сфальцьовані на контейнерах Docker, але уважно придивляємося й до іншим рішенням, що з'являються на ринку, включаючи Kubernetes і Mesos

АСІ і OpenStack

Аналітики прогнозують, що «підтримка OpenStack в АСІ повинна більш ніж просто задовольнити користувачів. Цей факт може взагалі усунути причини, по яких компанії розглядають продукти конкурентів».

Функціонал реалізується за допомогою OpFlex-агента, що підключається до Open vSwitch, того самим розширюючи дію політик АСІ до рівня гіпервізора. OpFlex – розроблена Cisco альтернатива протоколу OpenFlow, що у цьому

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

випадку забезпечує інтеграцію OpenStack і APIC-контролера. У результаті, користувачі OpenStack і ACI одержують повністю розподілений мережний стек Neutron, включаючи distributed-комутатори (аналог VMware vDS), маршрутизацію й NAT. Відновлення стане доступно в 4-м кварталі 2017 року. В останньому релізі ACI також використовує OpFlex для підтримки VMware vRealize, що використовується для керування віртуалізованою інфраструктурою, побудованої на базі VMware.

Інші відновлення

Крім вищеприведеного, у новий реліз увійшли наступні поліпшення (не повний список):

- Додаток в ACI Toolkit для автоматизації керування політиками в рамках декількох дата-центрів. ПЗ дозволяє синхронізувати політики між ЦОД для підвищення відкозостійкості й мобільності додатків;

- Підтримка SNMP і нового CLI для APIC, схожого з командним рядком NX-OS (підтримка запланована на 4-й квартал 2017 року).

Cisco позиціонує ACI як рішення для корпоративних клієнтів будь-якого масштабу, які шукають готове рішення від одного вендора. Компанія усвідомлює, що ACI – не для всіх, але робить ставку на те, що існуюча клієнтська база Cisco знайде її продукт більше прийнятним й зрозумілим, чим альтернативні SDN-рішення.

3.2 Розробка структурної схеми

Сучасні підходи до організації мережної інфраструктури центрів обробки даних у компанії Cisco одержали назву «інфраструктура, сфальцьована на додатках» (Application-Centric Infrastructure, ACI). У її основі – ACI-фабрика й керуючий контролер (Application Policy Infrastructure Controller, APIC). Концепція анонсована наприкінці минулого року й тепер стає комерційно доступною.

Ідея ACI дуже проста. Ми ставимо ідеологію інфраструктури ЦОД з голови на ноги: первинними елементами, на основі яких формуються правила

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

безпеки, обробки трафіку й т.д., є додатки. Все інше – спосіб реалізації цих правил. Даний підхід істотно відрізняється від традиційної мережної інфраструктури, де оперують такими поняттями як VLAN, підмережі й т.д., а потреби додатків намагаються привести до цих термінів за допомогою абстрагування ресурсів».

В АСІ базове поняття – мережний профіль додатка (див. Рисунок 3.1). У найпростішому випадку він описує правила взаємодії складовий додаток компонентів – його мережні й сервісні потреби. Міжмережні екрани, балансувальники навантаження й т.д. стають частиною опису інфраструктури додатка. Це опис повністю абстрагований від мережної й обчислювальної інфраструктури. Для нього не принципові кількість компонентів додатка і їхня природа (віртуальні вони або фізичні), IP-адресація, число серверів і т.д.

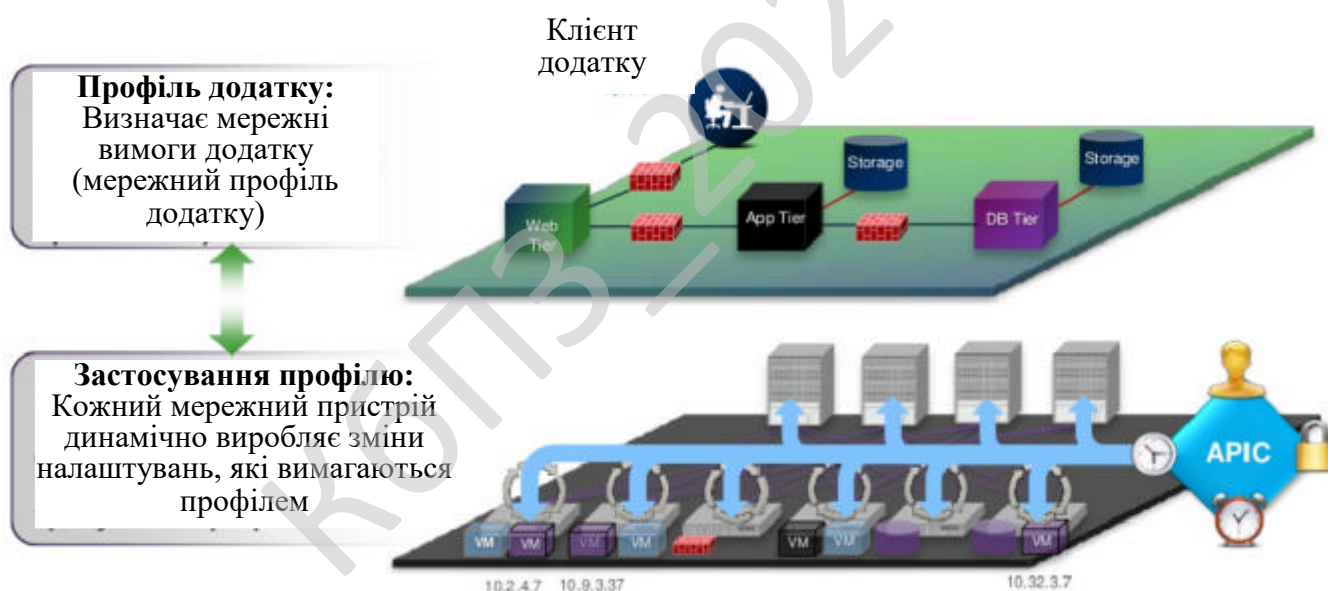


Рисунок 3.1 – Структурна схема системи

За допомогою профілів додатків можна управляти всією передачею даних у фабриці. Безпека й передача даних не залежать від фізичних і логічних мережних атрибутів. Адміністратор, відповідальний за додатки, формулює профіль додатка й передає такий опис контролеру APIC, керуючому всією

інфраструктурою. Воно доводить до всіх елементів мережі й фабрики. Всі подальші дії й зміни, такі як міграція віртуальних машин (VM) спрацьовуються фабрикою автономно й не вимагають додаткових керуючих впливів контролера, на відміну від SDN, де контролер повністю управляє комутаторами.

В Cisco вважають, що таке декларативне керування значно поліпшує масштабованість ACI. Задається лише логіка керування, але не його деталі. Можна додавати й нові мережні елементи, деталі логіки керування якими контролер знати не зобов'язаний. У той же час моделі на базі протоколу OpenFlow всі можливості мережі визначаються й контролером, і комутуючими пристроями. Технічно ніщо не заважає сполучити ACI і SDN, вони не виключають один одного. Але, на відміну від SDN з високою деталізацією централізованого керування в ACI застосовується декларативний підхід.

Мережні елементи самі підбудовуються під політики. В ACI кожний мережний пристрій динамічно застосовує налаштування відповідно до вимог профілю. По суті, APIC відіграє роль репозиторію політик, зберігає й поширює їх. Це єдина крапка керування для адміністратора, систем оркестрації й автоматизації, включаючи OpenStack, Cisco UCS Director і ін. Контролер вирішує також сервісні завдання, пов'язані з виявленням фабрики, відновленням ПЗ й т.д.

Висока масштабованість досягається також за рахунок того, що контролер не бере участь у долі кожного мережного пакета й перебуває за рамками передачі даних і змін, що відбуваються в мережі. Його завдання – поширювати й обновляти політики, збирати телеметричну інформацію, але він не програмує безпосередньо деталі комутації на кожному пристрої. У результаті кластер APIC здатний підтримувати більше мільйона кінцевих хостів, понад 200 тис. порти, 64 тис. логічних організацій (tenant).

Адміністратор може формулювати потреби додатка в термінах самого додатка. Описуються рівні додатків, їхня взаємодія без таких деталей як IP-адреси, VLAN та ін. Один раз сформулювавши політику, її можна використовувати надалі без зміни, наприклад, переносити між ЦОД, фізичними й

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

віртуальними серверами. Цей підхід працює й для повністю віртуалізованих, гібридних або не віртуалізованих середовищ.

Адміністратори додатків одержують детальну інформацію про те, що відбувається в мережі з його додатком – саме ту, котра відноситься до обслуговуючий додаток комутаторам у цей момент часу. Адміністратори безпеки мають своє бачення того, що відбувається в інфраструктурі ЦОД. Моніторинг подає інформацію про стан додатка й мережному трафіку всіх його компонентів. Для них ACI – це можливість керування правилами взаємодії компонентів інфраструктури на зрозумілому їм рівні. ACI підтримує улюблену ними модель «білого писку»: усе, що не дозволено, заборонено за замовчуванням. Крім того, можна вбудовувати додаткові засоби безпеки (Cisco або інших вендорів), віртуальні або фізичних, передбачені засоби аудита, протоколювання дій, API для зовнішнього аналізу».

Мережним адміністраторам ACI дає можливість піти від колишньої обробки сервісних заявок і налаштувань, експлуатувати мережну інфраструктуру не як набір пристроїв, а як комплекс із детальною телеметрією й діагностикою, ефективними механізмами перерозподілу трафіку між альтернативними маршрутами, підкреслюють в Cisco. Маршрутизація виконується в розподіленому режимі безпосередньо на границі мережі. При цьому можна розгортати середовища з довільним співвідношенням не віртуалізованих і віртуалізованих навантажень на різних платформах, впроваджувати гібридні й хмарні сервіси. Крім того, є можливість інтеграції з різними засобами оркестрації й автоматизації, включаючи OpenStack, чому Cisco приділяє особливу увагу. Опублікована модель даних і протокол OpFlex дозволяє додавати в ACI нові елементи, чому сприяє й відкритий механізм вбудовування сервісів L 4-L7.

Що стосується централізованого керування по політиках конвергентною інфраструктурою «у цілому», включаючи ПЗ віртуалізації, те дане завдання виконує Cisco UCS Director. Це один зі стратегічних продуктів Cisco, що заміняє кілька систем керування.

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

Як це працює? Адміністратори серверів, СЗД, мережі задають політики. Політики застосовуються для створення профілів серверів і конфігурацій LAN, SAN, СХД. Це дозволяє легко й швидко виконувати операції, що займають у реальній інфраструктурі досить багато часу. Процес максимально автоматизований.

З погляду архітектури USC Director – це одна або трохи ВМ, легко інтегровальні з існуючим фізичним і віртуальним середовищем: в USC Director просто вписується логін/пароль для спілкування з тим або іншим пристроєм. Система може містити в собі засобу розгортання ОС і гіпервізорів, підтримує інтеграцію із зовнішніми хмарами різних вендорів і різні сценарії використання.

Адміністратор може автоматизувати різні операції й застосовувати налаштування для надання сервісів користувачам. Платформа керування UCS Director відкрита нагору й долілиць: її можна інтегрувати з різним устаткуванням через спеціальні коннектори й SDK або викликати із зовнішніх систем. Ще одна можливість – інтеграція з білінговими системами для обліку споживаних ресурсів і виставлення рахунків.

Одна система UCS Director може управляти декількома «інтегрованими стеками», наприклад, vBlock і FlexPod, але крім конвергентних рішень Cisco підтримує серверні, мережні платформи й СЗД інших вендорів. У результаті можна вирішувати різні завдання по конфігуруванню інфраструктури, включаючи створення політик, профілів, пулів, шаблонів для Cisco UCS, створення томів і LUN, реєстрація ініціаторів та інші функції для дискових масивів EMC і NetApp, створення VLAN, налаштування конфігурації транкінгу й ін. у мережному встаткуванні, керування кластерами, віртуальними машинами й ін. у середовищах віртуалізації (VMware, Microsoft, RedHat).

Значна увага розроблювачі приділили автоматизації. Автоматизація дозволяє швидко виділяти ресурси й надавати сервіси ІТ на основі політик з контролем всіх операцій.

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

В Cisco планують і далі розвивати концепцію ACI і засобу керування. Це призначене для мережі ЦОД рішення може поширитися й за його межі – на магістральну й кампусну мережу, використовуватися в обчислювальних системах і системах зберігання).

3.3 Розробка функціональної схеми

На рисунку 3.2 зображена функціональна схема системи. Нижче розглянемо її більш докладно.

Система керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure призначена для рішення завдань диспетчеризації й моніторингу об'єктів у комп'ютерній мережі, автоматизованих за допомогою програмно-технічних засобів. Система керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure виконана в клієнт-серверній архітектурі. Серверна частина системи може бути встановлена на глобальному Інтернет-сервері або локальному сервері. Ніяких спеціальних програм на комп'ютері клієнта (користувача) встановлювати не потрібно, досить мати лише Інтернет-браузер і вихід у мережу Інтернет або локальна мережа, залежно від побудови загальної системи моніторингу.

Короткий опис основних функціональних блоків системи.

Розробка мнемосхем у середовищі розробленого програмного забезпечення системи збору, обробки та відображення інформації об'єкту моніторингу у мережі

Розробка мнемосхем може бути здійснена безпосередньо в редакторі керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure. Дана можливість опирається на бібліотеку примітивів, які мають ряд параметрів. Параметри «за замовчуванням» можуть мінятися користувачем. Середовище дозволяє створювати мнемосхеми, які будуть виконувати закладений у них функціонал. Параметри (характеристики, значення) одного примітива

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

можуть впливати на параметри іншого, у результаті утвориться повнофункціональна схема із залежними елементами. Дана можливість обумовлена динамічними властивостями примітивів. У результаті створюється деякий сценарій, відповідно до якого функціонує вся система, або її блоки. У загальному бібліотеку примітивів можна розділити на кілька груп: фігури (або об'єкти) використовувані в технологічному процесі (баки, казани, труби та ін.), датчики, логіка, арифметика й т.п. У цілому є каталог примітивів, розбитий у групи. Є можливість створення нової групи примітивів, редагування наявної й т.п.

Розробка примітивів

Можливість створення власних примітивів, з додаванням їх у базу наявних. Дана можливість дозволить створювати примітиви із графічних зображень, у тому числі анімованих; з найпростіших фігур: прямокутників, ліній і т.п. Такий примітив буде наділений властивостями (параметрами) і може бути використаний як повноцінний примітив, що має «розум». При створенні він міститься в зазначену групу примітивів. Таким чином, здійснюється нарощування бібліотеки. Бібліотека примітивів виконується як окремий функціональний модуль, що може бути оновлений, не залежно від системи в цілому. Можливе створення спеціалізованих бібліотек примітивів для різних технологічних процесів.

Спостереження в реальному часі мнемосхем об'єктів моніторингу з параметрами, що змінюються динамічно

Дана можливість розроблена з використанням динамічного завантаження (підзавантаження) даних. Достоїнством даної технології є те, що немає необхідності багаторазово обновляти дані на сторінці. На завантаженої один раз мнемосхемі відбувається відновлення лише необхідної інформації: параметрів (значень), завантаження потрібних зображень і т.п. При цьому вся інша сторінка вже не завантажується. Система дозволяє міняти не тільки значення у вигляді цифр, рядків, але й змінювати форму, колір, положення примітива. Оскільки дана

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

система використовується в мережі Інтернет, то дозволяє раціонально використовувати трафік, виконуючи лише необхідні відновлення на сторінці.

Управління встаткуванням

Оскільки система являє собою комплекс пристроїв і програмного забезпечення, тобто можливість не тільки спостерігати за параметрами мнемосхеми, але й вносити нові значення. Це дає можливість управляти Технологічним Процесом. Перш ніж значення будуть внесені (змінені), вони пройдуть перевірку на валідність (правильність) і адекватність. Це забезпечується заданими обмеженнями на значення.

Графічний моніторинг об'єкту моніторингу

Для наочного подання процесу, що відбувається, використовується графічний моніторинг. На графіку більш наочно можна бачити зміни й статистику роботи різних пристроїв, блоків, реєстрації датчиків і т.п.

Засоби оповіщення

Для найбільш швидкого реагування на різні ситуації, що відбуваються в мережі, використовуються засоби оповіщення: SMS-повідомлень і відправлення електронної пошти. Це може бути корисним при відмовах системи, помилках і інших позаштатних ситуаціях. Також можна одержувати статистику від системи поштою за певний період її роботи, якщо така необхідність є.

Планувальник

У систему закладена можливість задавати параметри системи на певний час (години : хвилини день/місяць/рік). Може задаватися інтервал, на якому необхідно мати певні параметри. У результаті система працює згідно закладеного в неї плану (програми).

Система логістики

Система веде логи (логістику) по всьому технологічному процесі. У користувачів є можливість у будь-який момент відобразити ці параметри на екрані (або вивести їх в xls-файл). По певних характеристиках можна побудувати

графік для наочного моніторингу процесу. Система логістики фіксує всі зміни, які вносить кожний користувач, тобто всі дії фіксуються.

Права доступу до системи

Передбачений 3-х рівневий (SuperAdmin, Admin і User) багатокористувальницький доступ користувачів у систему. Права на групи Admin і User визначаються при реєстрації користувачів. Кожна група має певні можливості й доступ до розділів системи.

Відображення структури Об'єкту моніторингу, максимально наближеної до реальності

Створюється мнемосхема, яка копіює технологічний процес, з усіма можливими елементами які в ньому присутні. Іншими словами це просто модель процесу. Число мнемосхем у проекті не обмежено.

Тобто існує можливість продублювати деякі Об'єкти моніторингу в одному проекті, або ж попросту розбити один процес на більше дрібні підпроцеси.

Число елементів мнемосхем не обмежено. Тут варто керуватися тільки наочністю процесу. Тобто якщо у вас багато елементів на схемі то попросту можна не зрозуміти де що – плутанина. Як елемент мнемосхеми може бути використаний будь-який елемент управління ActiveX, з можливістю динамізації будь-якої його властивості, а також одного зі стандартних властивостей (положення, розміру, відрисовки, миготіння й т.п.).

В основі створення мнемосхем лежить використання стандартних елементів, типових. Бібліотеки типових елементів нараховують біля тридцяти стандартних елементів, включаючи об'ємні елементи з убудованим індикатором заповнення, елементи для створення користувальницьких діалогів, елементи, що відтворюють повний комплект приладів щитового контролю й управління. Є убудований редактор для створення мультфільмів (з регульованою прозорістю зображення) з різними законами трансформації вихідних графічних файлів (покадровий показ, прокручування в будь-якому напрямку, зміна різкості або

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

розміру й т.п. із завданням часу й кількості кадрів). Об'ємні трубопроводи довільної конфігурації створюються в кілька клацань миші.

Існує можливість використання в мнемосхемі таких елементів як gif-анімація, різних відео файлів.

Підтримуються всі стандартні графічні формати: bmp, gif, jpg, avi. Всі імпортовані зображення й відеокліпи можуть бути відображені в режимі з прозорістю, що налаштовується, і одночасної динамізації будь-яких інших властивостей.

Так само використовується векторна графіка, що дає широке коло операцій над обраним об'єктом.

Аналіз архіву параметрів Об'єкту моніторингу в графічному виді

Для аналізу й подання інформації використовуються тренди. Тренди призначені для перегляду даних у графічному й табличному виді.

Можливе відображення будь-якого числа графіків будь-яких архівуємих і неархівуємих змінних. Вставка змінних у тренд виробляється їхнім перетаскуванням із проекту з автоматичним спадкуванням діапазону й одиниці виміру. Тренд реального часу й історичний об'єднаний в одному вікні. Кожне перо може мати свій діапазон осі значень. Масштаб часу й значень може бути змінений у процесі перегляду.

Існує можливість зберегти тренд у вигляді графічного файлу з розширенням jpg у папку "Тренди" даного об'єкту.

Створення звітів і їх друк в призначений час

Для створення рапортів (кількість рапортів не обмежене) використовується Microsoft Excel (надалі буде також можливо використовувати убудований редактор рапортів). Excel відкривається безпосередньо у вікні редагування пакета. Змінні в таблицю рапорту перетаскуються з дерева проекту. Вставлені змінні можна використовувати у формулах графіках і діаграмах стандартним образом. Друк або збереження рапортів виробляється за розкладом або подією в зручний час.

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

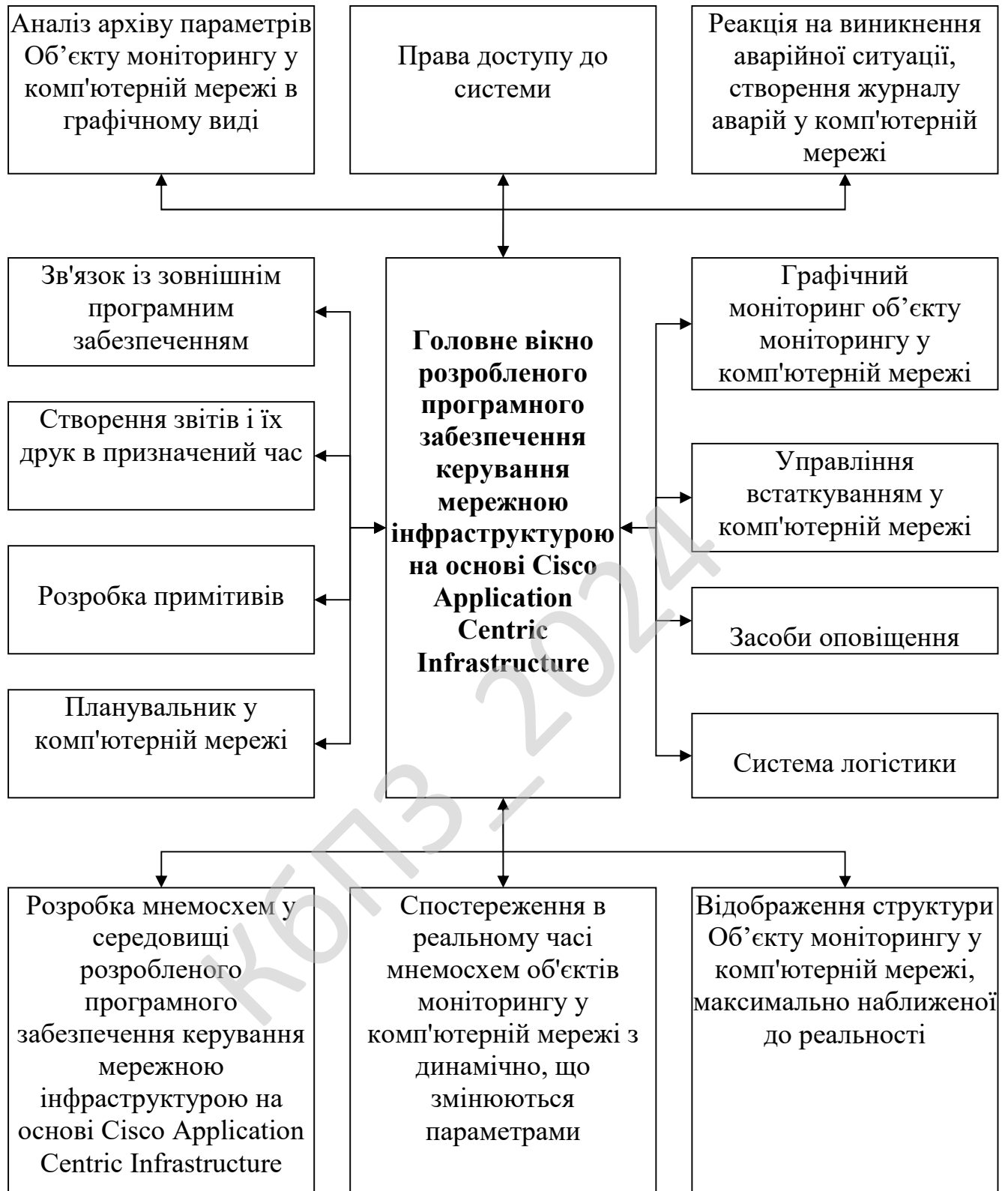


Рисунок 3.2 – Функціональна схема системи

Повідомлення завжди відноситься до тієї або іншої Категорії. Є ряд визначених категорій. Є можливість створення своїх категорій.

Категорія – атрибут, що дозволяє сортувати й фільтрувати повідомлення залежно від їхнього призначення.

Категорія має ім'я й налаштування:

- колір тексту й тла повідомлень;
- звуковий файл, що програватиметься при виникненні повідомлення;
- пріоритет;
- список каналів виводу (для каналу виводу Журнал задається перелік журналів, у які будуть надходити повідомлення даної категорії);

Основний спосіб перегляду повідомлень, включаючи архівні, це журнал. Журнал повідомлень є документом, що належить об'єкту. У ньому передбачені засоби фільтрації й сортування по кожному з полів і/або джерел повідомлень. Журнал може бути роздрукований, експортований у вигляді XML-файлу або опублікований в Інтранеті/Інтернеті. Журнал є клієнтом повідомлень у стандарті OPC.

Журнал – засіб для перегляду повідомлень, у якому реалізована можливість фільтрувати повідомлення по тим або інших ознаках. Журнал відкривається зі сторінок властивостей об'єкту (а також за розкладом об'єкту або по дії, налаштованій для події цього об'єкту). Можна створити кілька форматів Журналів, що мають різний зовнішній вигляд, заголовки, фільтри для повідомлень і т.д. Приналежність Журналу до того або іншого Об'єкту визначається лише тим, що в ньому видні повідомлення тільки від джерел, що належать даному Об'єкту. Журнал не є в точному значенні документом об'єкту, тому що не зберігає повідомлення, а тільки виводить їх на екран.

Журнали об'єктів можуть відкриватися як з режиму розробки, так і з режиму виконання. При відкритті Журналу в режимі розробки переглядати можна тільки архівні повідомлення (кнопка включення фільтра активних подій у цьому випадку недоступна). Для перегляду вмісту Журналу Можливо

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

використовувати додаткові фільтри, за допомогою яких можна внести заборони на вивід повідомлень по тих або інших ознаках. Таким чином, відкриваючи який-небудь Журнал об'єкту, можливо бачити вибірку повідомлень, зроблену виходячи з наступних позицій:

- від елементів тільки цього об'єкту й, можливо, дочірніх;
- відповідно до налаштованого для даного типу Журналу фільтрами;
- відповідно до налаштувань додаткових фільтрів.

Фільтрація повідомлень по категоріях відбувається зворотним образом: у налаштуваннях кожної категорії вказується, у яких Журналах відобразити повідомлення даної категорії. За замовчуванням для кожної категорії вказується тільки "Основний журнал", інші типи Журналів варто призначити спеціально.

Зв'язок із зовнішнім програмним забезпеченням

Можливість експорту архіву параметрів. Підтримуються архіви даних, повідомлень і рапортів. Обсяги архівів обмежуються тільки самим користувачем.

Архіви розподілені по об'єктах. Інформація в архів даних направляється з використанням індивідуально обраного для кожної змінної фільтра. Можна задати обмеження тривалості або обсягу зберігання індивідуально для кожного об'єкту й типу архіву. Для сумісності із зовнішніми базами даних забезпечується експорт архівів (у тому числі в режимі off-line) з можливістю об'єднання архівів декількох об'єктів в один загальний зовнішній архів.

Для перегляду Архіву даних його можна експортувати у файл *.mdb (після чого за допомогою Microsoft Access його можна переглянути або конвертувати в зручний для Вас формат). Швидко переглянути дані ви можна так само на закладці "Дані" в змінній, яку необхідно переглянути, (перегляд можливий, якщо в змінній встановлений прапор "Архівувати"). Експорт Архівів даних або повідомлень може здійснюватися автоматично, за розкладом Об'єкту, або по дії Події.

Існує можливість у проектах використовувати MS SQL, що у свою чергу дозволяє створювати незалежні програми для аналізу інформації,

використовувати вже наявне програмне забезпечення, орієнтоване на обробку даних.

Можливість перегляду даних через Internet. Доступ з будь-якого комп'ютера підприємства до інформації, що надходить від виробничого об'єкту моніторингу, від будь-якої підсистеми стає необхідністю. А різного типу клієнтські додатки можуть надавати відповідному виробничому процесу у величезному обсязі дані в прийнятному для користувача виді.

Тому існує можливість перегляду всіх документів і властиво мнемосхем через Internet.

Зв'язок з нижнім рівнем – Контролером. Зв'язок з нижнім рівнем здійснюється по засобах ОПС сервера, що здійснює буферизацію даних які йдуть від фізичного пристрою, щоб потім розподілити їх між різними клієнтськими додатками або управляє передачею даних на різні фізичні пристрої, по запитах відповідних клієнтських додатків.

ОПС дає можливість у користувальницькі додатки інтегрувати нові частини програмного забезпечення.

Постачальники ПЗ промислового призначення можуть розробляти продукти, сумісні з розповсюдженими стандартами ПЗ.

Таки образом, підвищується продуктивність і якість продукту.

Можливість контролю дій оператора й обмеження прав доступу. Є розвинені засоби багаторівневого обмеження прав доступу, що включають формування переліку посад з налаштуванням прав доступу для кожної посади, адміністрування призначення операторів на посаді й у зміни, реєстрацію в зашифрованому журналі всіх дій операторів.

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

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

3.4 Розробка діаграми процесів

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

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

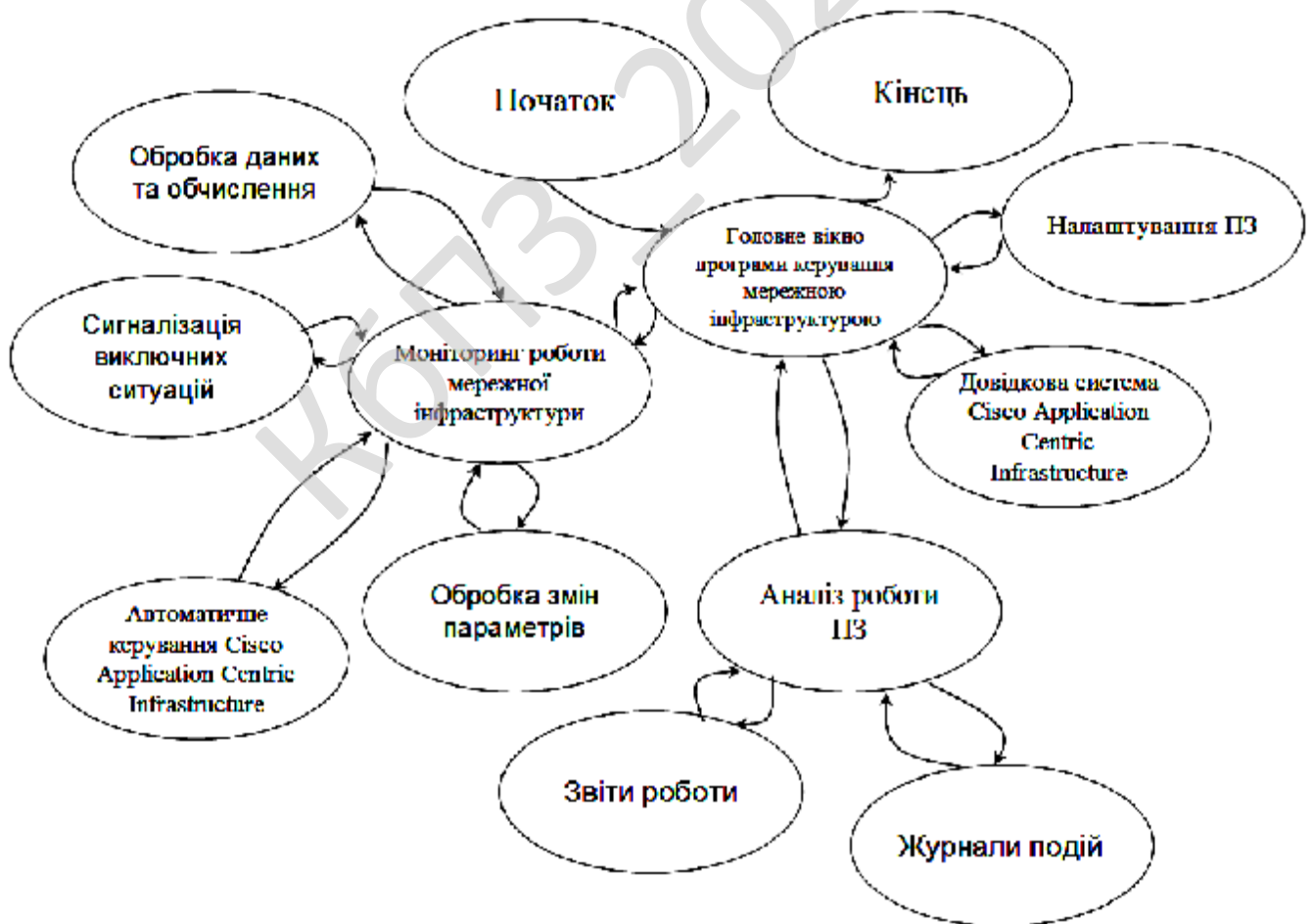


Рисунок 3.3 – Діаграма взаємодії процесів

Діаграми потоків даних містять чотири типи елементів:

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

– Сховища даних (репозиторії).

– Зовнішні по відношенню до системи сутності.

– Потоки даних між елементами трьох попередніх типів.

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

КБПЗ_2024

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

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

При виборі початкової точки відліку при побудові схем було враховано, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об'єктно-орієнтована що вимагає оптимізації, також те, що при розробці програми слід надати особливу увагу модулю керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure.

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

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

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

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

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

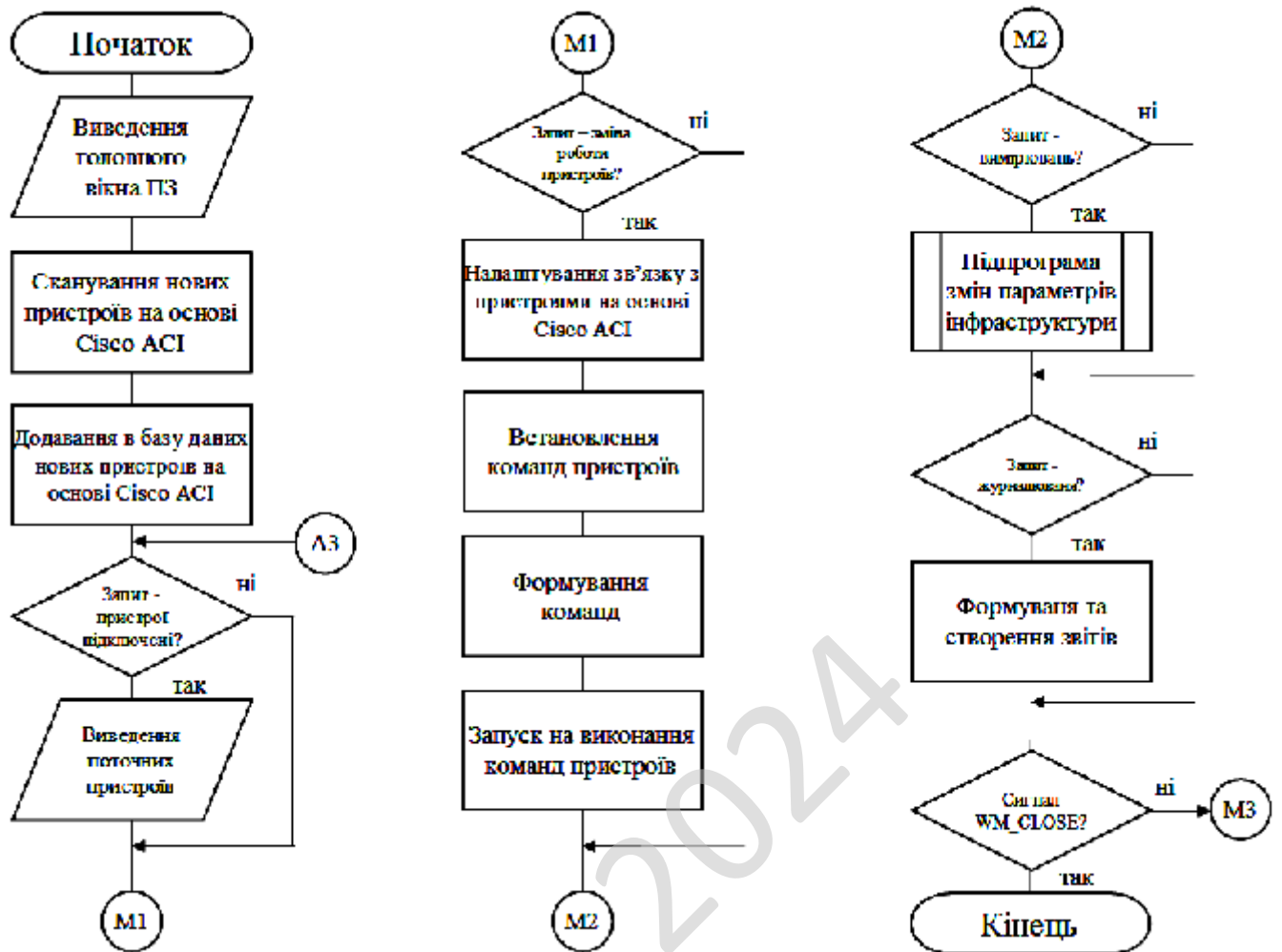


Рисунок 4.1 – Блок-схема основної програми

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

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

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

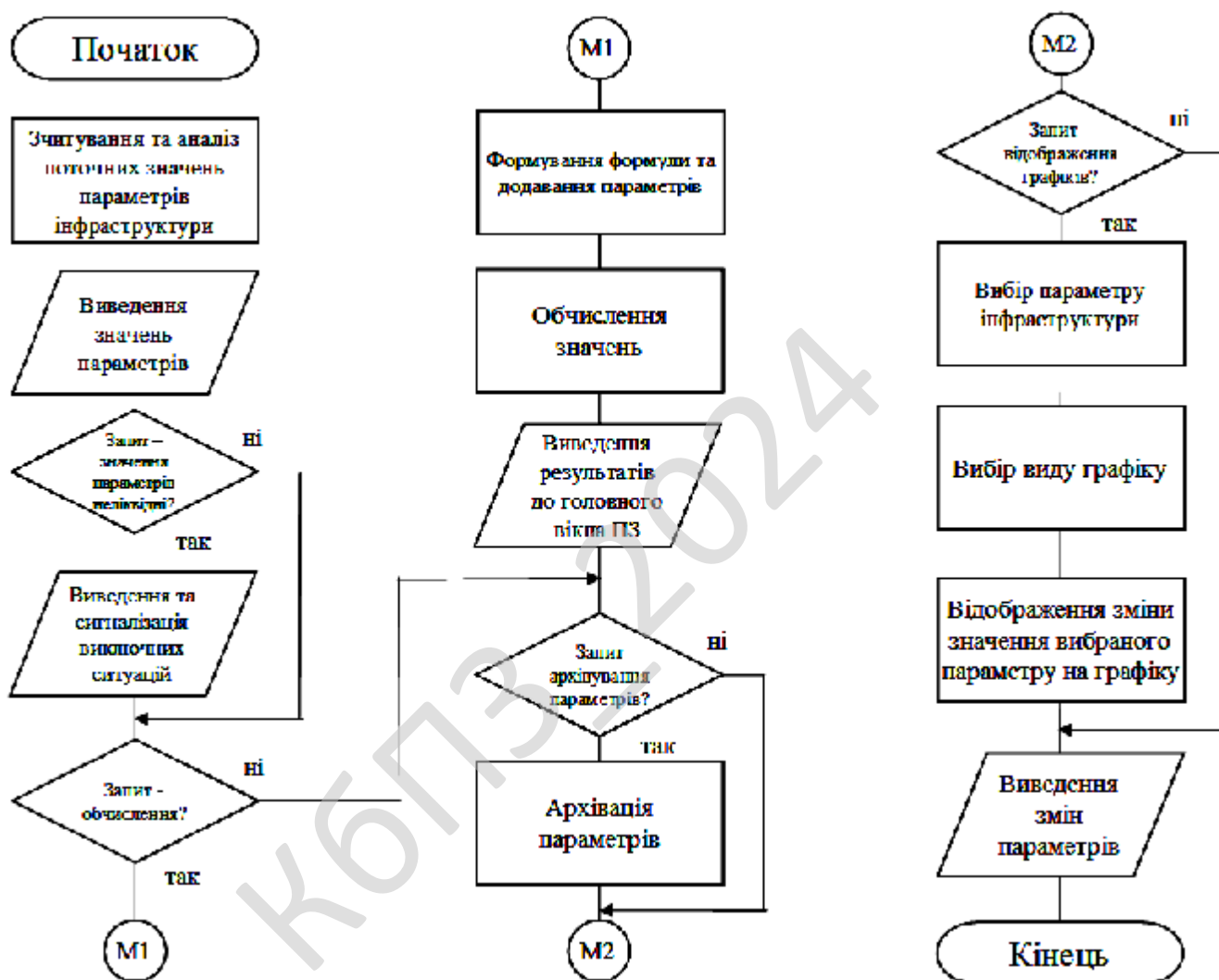


Рисунок 4.2 – Блок-схема роботи підпрограми

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

Redmine – вільне серверне ПЗ для управління проектами та відстежування помилок. До системи входить календар-планувальник та діаграми Ганта для візуального представлення ходу робіт за проектом та строків виконання. Redmine

написано на мові Ruby і є ПЗ розробленим з використанням відомого веб-фреймворку Ruby on Rails, що означає легкість в розгортанні системи та її адаптації під конкретні вимоги. Для кожного проекту можна вести свої вікі та форуми.

Функціональні можливості:

- Ведення декількох проектів.
- Гнучка система доступу з використанням ролей.
- Система відстеження помилок.
- Діаграми Ганта та календар.
- Ведення новин проекту, документів та управління файлами.
- Сповіщення про зміни за допомогою RSS-потоків та електронної пошти.
- Власна Wiki для кожного проекту.
- Форуми для кожного проекту.
- Облік часових витрат.
- Налаштування власних (custom) полів для задач, затрат часу, проектів та користувачів.

– Легка інтеграція із системами керування версіями (SVN, CVS, Git, Mercurial, Vazaar и Darcs).

- Створення записів про помилки на основі отриманих листів
- Підтримка LDAP автентифікації.
- Можливість самореєстрації нових користувачів.
- Багатомовний інтерфейс (у тому числі українська мова).
- Підтримка СКБД: MySQL, PostgreSQL, SQLite.

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

Діаграма Ганта являє собою відрізки (графічні плашки), розміщені на горизонтальній шкалі часу. Кожен відрізок відповідає окремому завданню або

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

підзадачі. Завдання і підзадачі, складові плану, розміщуються по вертикалі. Початок, кінець і довжина відрізка на шкалі часу відповідають початку, кінцю і тривалості завдання. На деяких діаграмах Ганта також показується залежність між завданнями.

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

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

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

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

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

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

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

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

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

- номер (ідентифікатор) дефекту;
- хто повідомив про дефект;
- дата і час виявлення дефекту;
- версія продукту, в якій виявлено дефект;
- серйозність (критичність) дефекту та пріоритет рішення;
- опис кроків для відтворення дефекту (неправильної поведінки програми);
- відповідальний за усунення дефекту;
- обговорення можливих рішень та їх наслідків;
- поточний стан виправлення дефекту;
- версії продукту, в якій дефект виправлений.

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

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

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

Типовий життєвий цикл дефекту:

1. Новий – дефект зареєстрований тестувальником.
2. Призначений – призначений відповідальний за виправлення дефекту.

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

3. Дозволений – дефект переходить назад у сферу відповідальності тестувальника. Як правило, супроводжується резолюцією, наприклад:

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

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

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

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

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

Основна частина системи це – модулі. Тісна інтеграція модулів з ядром поліпшує стабільність системи в цілому, завдяки повторному використанню налагодженого коду.

Загальносистемні користувальницькі об'єкти. Абстрактний об'єкт являє собою асоціативний контейнер властивостей і функцій. Властивості можуть містити як дані чотирьох базових типів, так і інші об'єкти.

Доступ до властивостей об'єкта звичайно здійснюється за допомогою запису імен властивостей через крапку до об'єкта `<obj.prop>`, а також за допомогою висновку ім'я властивості у квадратні дужки `<obj["prop"]>`.

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

Очевидно, що перший механізм статичний, а другий дозволяє вказувати ім'я властивості через змінну. Базове визначення об'єкта не містить функцій. Операції копіювання об'єкта насправді роблять посилання на вихідний об'єкт. При видаленні об'єкта здійснюється зменшення лічильника посилань, а при досягненні лічильника посилань нуля об'єкт віддаляється фізично.

Різні компоненти можуть довізначати базовий об'єкт особливими властивостями й функціями. Стандартним розширенням об'єкта є масив "Array".

Об'єкт Array. Особливістю масиву є те, що він працює із властивостями, як з індексами, і повне їхнє іменування безглуздо, а значить доступний механізм обігу тільки висновком індексу у квадратні дужки <arr[1]>. Масив зберігає властивості у власному контейнері одномірного масиву. Цифрові властивості масиву використовуються для доступу безпосередньо до масиву, а символічні працюють як властивості об'єкта.

Масив надає спеціальну властивість "length" для одержання розміру масиву <var = arr.length;>. Також масив надає наступні функції:

– string join(string sep = ","), string toString(string sep = ","), string valueOf(string sep = ",") – Повертає рядок з елементами масиву розділеними <sep> або символом

– Array concat(Array arr); – Додає до вихідного масиву елементи масиву <arr>. Повертає вихідний масив зі змінами.

– int push(EType var, ...); – Поміщає елемент(ы) <var> у кінець масиву, як у стек. Повертає новий розмір масиву.

– EType pop(); – Видалення останнього елемента масиву й повернення його значення, як зі стека.

– Array reverse(); – Зміна порядку розташування елементів масиву. Вертається вихідний масив зі змінами.

– EType shift(); – Зрушення масиву у верх. При цьому перший елемент масиву віддаляється, а його значення вертається.

– `int unshift(ElTp var, ...)`; – Засовує елемент(и) `<var>` у масив. Перший елемент в 0, другий в 1 і т.д.

– `Array slice(int beg, int end)`; – Повертає фрагмент масиву від `<beg>` до `<end>`. Якщо значення початку або кінця негативно, то відлік ведеться з кінця масиву. Якщо кінець не зазначений, то кінцем є кінець масиву.

– `Array splice(int beg, int rem, ElTp val1, ElTp val2, ...)`; – Вставляє, видаляє або заміняє елементи масиву. Повертає вихідний масив зі змінами. У першу чергу здійснюється видалення елементів з позиції `<beg>` і кількістю `<rem>`, а потім вставляються значення `<val1>` і т.д., починаючи з позиції `<beg>`.

– `Array sort()`; – Сортуювання елементів масиву в лексикографічному порядку.

Об'єкт `RegExp`. Об'єкт роботи з регулярними вираженнями, заснований на бібліотеці `PCRE`. При глобальному пошуку встановлюється атрибут об'єкта `"lastIndex"`, що дозволяє продовжити пошук при наступному виклику функції. У випадку невдалого пошуку атрибут `"lastIndex"` скидається в нуль.

Як аргументи створення об'єкта передається рядок з текстом регулярного вираження й прапори у вигляді рядка символів:

– `'g'` – режим глобального пошуку;

– `'i'` – режим регістронезалежного пошуку;

– `'m'` – режим багаторядкового пошуку;

– `'r'` – тестування вираження по звичайному шаблоні із ключовими символами: `'?'`, `'*'` і `'\'`.

Властивості об'єкта:

– `source` – Вихідний шаблон регулярного вираження, тільки читання.

– `global` – Ознака глобального пошуку, тільки читання.

– `ignoreCase` – Ознака ігнорувати регістр символів при пошуку, тільки читання.

– `multiline` – Ознака багаторядкового пошуку, тільки читання.

– int time(int usec); – повертає абсолютний час у секундах від епохи 1.1.1970 і мікросекундах, якщо <usec> зазначений.

– int localtime(int fullsec, int sec, int min, int hour, int mday, int month, int year, int wday, int yday, int isdst); – повертає повну дату в секундах (sec), хвиликах (min), годинниках (hour), днях місяця (mday), місяці (month), році (year), днях тижня (wday), днях у році (yday) і ознака літнього часу (isdst), виходячи з абсолютного часу в секундах <fullsec> від епохи 1.1.1970.

– string strftime(int sec, string form = "% Y-Y-%m-%d %H:%M:%S"); – Перетворить абсолютний час <sec> у рядок потрібного формату <form>. Запис формату відповідає POSIX-функції strftime.

– int strptime(string str, string form = "% Y-Y-%m-%d %H:%M:%S"); – Повертає час у секундах від епохи 1.1.1970, виходячи зі строкового запису часу <str>, відповідно до зазначеного шаблону <form>. Наприклад, шаблону "% Y-Y-%m-%d %H:%M:%S" відповідає час " 2006-08-08 11:21:55". Опис формату шаблону можна одержати з документації на POSIX-функцію "strptime".

– int cron(string cronreq, int base = 0); – Повертає час, спланований у форматі стандарту Cron <cronreq>, починаючи від базового часу <base> або від поточного, якщо базове не зазначено.

– string strFromCharCode(int char1, int char2, int char3, ...); – Створення рядка з кодів символів char1, char2 ... char.

– string strCodeConv(string src, stringfromCP, string toCP); – Кодування тексту <src> з кодування <fromCP> в <toCP>. Якщо кодування опущене, то використовується внутрішня.

Будь-який об'єкт (TCntrNode) дерева WebSCADA (SYS.*). Функції об'єкта:

– TArrayObj nodeList(string grp = "", string path = ""); – Одержання списку дочірніх вузлів для групи <grp> і вузла по шляху <path>. Якщо <grp> порожня то вертаються вузли всіх груп.

– TCntrNodeObj nodeAt(string path, string sep=""); – Підключення до вузла <path> у дереві об'єктів WebSCADA. Якщо вказується роздільник в <sep> то шлях обробляється як рядок з роздільником.

– TCntrNodeObj nodePrev(); – Одержати попередній, батьківський, вузол.

– string nodePath(string sep = "", bool fromroot = true); – Одержання шляху до поточного вузла, у дереві об'єктів WebSCADA. Один символ роздільника вказується в <sep> для одержання шляху через роздільник, наприклад, "DAQ.ModBus.PLC1.P1.var", інакше "/DAQ/ModBus/PLC1/P1/var". <from_root> указує на необхідність формувати шлях від кореня й без вказівки ідентифікатора станції.

Підсистема "Безпека" (SYS.Security). Функції об'єкта підсистеми (SYS.Security):

– int access(string user, int mode, string owner, string group, int access) – Перевірка для користувача <user> доступу до ресурсу який належить <owner> і <group> з доступом <access> для режим <mode>:

– user – користувач для перевірки доступу; mode – режим доступу (4-R, 2-W, 1-X); owner – власник ресурсу; group – група ресурсу;

– access – режим доступу до ресурсу (RWXRWXRWX – 0777).

Функції об'єкта користувача (SYS.Security["usr_User"]) і групи (SYS.Security["grp_Group"]):

– ElTp cfg(string nm) – одержання значення конфігураційного поля <nm> об'єкта.

– ElTp cfgSet(string nm, ElTp val) – установка конфігураційного поля <nm> об'єкта в значення <val>.

Підсистема "БД" (SYS.BD).

Функції об'єкта БД (SYS.BD["TypeDB"]["DB"]):

– ElTp cfg(string nm) – одержання значення конфігураційного поля <nm> об'єкта.


```
SYS.BD.MySQL.GenDB.SYS.fieldGet (req);
```

```
SYS.messDebug ("TEST DB", "Value: "+req.childGet (2).text ());
```

– string fieldSet(XMLNodeObj fld); – Установка поля. У випадку помилки вертається "0:Error".

– string fieldDel(XMLNodeObj fld); – Видалення поля. У випадку помилки вертається "0:Error".

Підсистема "Збір даних" (SYS.DAQ). Функції об'єкта підсистеми (SYS.DAQ):

– bool funcCall(string progLang, TVarObj args, string prog); – виклик тексту функції <prog> з аргументами в об'єкті <args> для мови програмування <progLang>. Повертає "true" при коректному виклику.

```
var args = new Object (); args.y = 0; args.x = 123;  
SYS.DAQ.funcCall ("BuilderCLikeCalc.BuilderCScript", args, "y=2*x;");  
SYS.messDebug ("TEST Calc", "TEST Calc result: "+args.y);
```

Була використана водоспадна (каскадна) модель життєвого циклу ПЗ (waterfall model) – послідовний метод розробки програмного забезпечення, названий так через діаграму схожу на водоспад.

Ця модель розробки запозичена з системної інженерії у виробництві та будівництві – областях, в яких зміни на пізніх етапах дуже дорогі, або неможливі. Наприклад, для створення складних інженерних конструкцій (споруд, літаків, мостів і т.п.). Зміни в проекті фундаменту будинку після того, як покладений дах коштують дуже дорого, тому перфекціонізм на початкових етапах проектування просто необхідний. Інженери, які починали займатись розробкою програмного забезпечення перейшовши з інших галузей, просто адаптували звичну модель, тому що на ранніх етапах розвитку комп'ютерної техніки не було методологій створених саме для програмування. Проте, схожі методології застосовуються для програмного забезпечення й далі, у випадках коли вимоги фіксовані, і вимагається висока якість та надійність, наприклад в системах для військових чи медичних потреб.

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

Перший формальний опис водоспадної моделі, після якої вона стала популярною був здійснений В. В. Ройсом у 1970. Попри те, що стаття містить переважно критику методу, на неї часто посилаються.

Переваги методу:

- Ніяких переробок.
- Гарна специфікація перетікає в гарну документацію.
- Зрозуміла модель.
- Розробники можуть мати низьку кваліфікацію.

Недоліки:

- Необхідний перфекціонізм на кожному етапі.
- Важко вносити зміни (якщо взагалі можливо).
- Надлишкове проектування.
- Поділ розробників на "perfect" та "code monkeys".

Модифікації. Через те що цей метод погано підходить для розробки саме ПЗ, частіше використовують його модифікації.

Найвідоміша модифікація – Sashimi. Названа так через японську страву сашімі (суші нарізане і сервіроване так, що складені рядочком шматочки накладаються один на одного). В моделі розробки Сашімі фази життєвого циклу йдуть одна за одною, але при цьому перекриваються одна з одною в часі.

4.2 Захист розробленого програмного забезпечення

Розроблене програмне забезпечення захистимо за допомогою національного стандарту захисту інформації на основі алгоритму шифрування/дешифрування ДСТУ 4145-2002 з використанням еліптичних кривих над двійковим розширеним полем Галуа. У системі шифрування/дешифрування як параметри розглядається еліптична крива $E_p(a,b)$ і точка G на ній. Учасник B вибирає закритий ключ n і обчислює відкритий ключ $P_B = n \times G$. Щоб зашифрувати повідомлення P_m використовується відкритий ключ

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

одержувача В P_B . Учасник А вибирає випадкове ціле позитивне число k і обчислює зашифроване повідомлення C_m , що є точкою на еліптичній кривій.

$$C_m = \{k \times G, P_m + k \times P_B\}. \quad (4.1)$$

Щоб дешифрувати повідомлення, учасник В множить першу координату точки на свій закритий ключ і віднімає результат від другої координати:

$$P_m + k \times P_B - n_B \times (k \times G) = P_m + k \times (n_B \times G) - n_B \times (k \times G) = P_m. \quad (4.2)$$

Учасник А зашифрував повідомлення P_m додаванням до нього $k \times P_B$. Ніхто не знає значення k , тому, хоча P_B і є відкритим ключем, ніхто не знає $k \times P_B$. Супротивнику для відновлення повідомлення доведеться обчислити k . Зробити це буде нелегко. Одержувач також не знає k , але йому як підказку посилається $k \times G$. Помноживши $k \times G$ на свій закритий ключ, одержувач одержить значення, що було додано відправником до незашифрованого повідомлення. Тим самим одержувач, не знаючи k , але маючи свій закритий ключ, може відновити незашифроване повідомлення.

Нехай задано просте число $p > 4$. Тоді еліптичною кривою E , визначеною над розширеним двійковим полем F_{2^m} , називається безліч пар чисел (x, y) , $x, y \in F$, що задовольняють тотожності:

$$y^2 \equiv x^3 + a \cdot x + b \pmod{2^m}, \quad (4.3)$$

де $4 \cdot a^3 + 27 \cdot b^2$ не рівно з нулю по модулю 2^m .

Інваріантом еліптичної кривої називається величина $J(E)$, що задовольняє тотожності:

$$J(E) \equiv 1728 \frac{4a^3}{4a^3 + 27b^2} \pmod{2^m}. \quad (4.4)$$

Коефіцієнти a , b еліптичної кривої E , по відомому інваріанту $J(E)$, визначаються таким чином:

$$\begin{cases} a \equiv 3k \pmod{2^m} \\ b \equiv 2k \pmod{2^m} \end{cases} \text{де } k \equiv \frac{J(E)}{1728 - J(E)} \pmod{2^m}, J(E) \neq 0 \text{ або } 1728. \quad (4.5)$$

Пари (x, y) , що задовольняють тотожності (4.1), називаються точками еліптичної кривої E , x та y – відповідно x - та y -координатами точки.

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

Точки еліптичної кривої позначатимемо $Q(x,y)$ або просто Q . Дві точки еліптичної кривої рівні, якщо рівні їх відповідні x - і y -координати.

На безлічі всіх точок еліптичною кривою E введемо операцію додавання, яку позначатимемо знаком "+". Для двох довільних точок $Q_1(x_1, y_1)$ та $Q_2(x_2, y_2)$ еліптичної кривої E , розглянемо декілька варіантів.

Нехай координати точок Q_1 та Q_2 задовольняють умові $x_1 \neq x_2$. В цьому випадку їх сумою називатимемо точку $Q_3(x_3, y_3)$ координати якої визначаються порівняннями:

$$\begin{cases} x_3 \equiv \lambda^2 - x_1 - x_2 \pmod{2^m}, \\ y_3 \equiv \lambda(x_1 - x_3) - y_1 \pmod{2^m}, \end{cases} \text{ де } \lambda \equiv \frac{y_2 - y_1}{x_2 - x_1} \pmod{2^m}. \quad (4.6)$$

Якщо виконана рівність $x_1 = x_2$ та $y_1 = y_2 \neq 0$, то визначимо координати точки Q_3 таким чином:

$$\begin{cases} x_3 \equiv \lambda^2 - 2x_1 \pmod{2^m}, \\ y_3 \equiv \lambda(x_1 - x_3) - y_1 \pmod{2^m}, \end{cases} \text{ де } \lambda \equiv \frac{3x_1^2 + a}{2y_1} \pmod{2^m}. \quad (4.7)$$

У разі, коли виконана умова $x_1 = x_2$ та $y_1 = -y_2 \pmod{p}$, суму точок Q_1 та Q_2 називатимемо нульовою точкою O , не визначаючи її x - і y -координати. В цьому випадку, точка Q_2 називається запереченням точки Q_1 . Для нульової точки O виконана рівність:

$$Q + 0 = 0 + Q = Q, \quad (4.8)$$

де Q – довільна точка еліптичної кривої E .

Щодо введеної операції складання безліч всіх точок еліптичною кривою E , разом з нульовою точкою, утворюють кінцеву абельову (комутативну) групу порядку t , для якого виконана нерівність:

$$p + 1 - 2\sqrt{p} \leq t \leq p + 1 + 2\sqrt{p} \quad (4.9)$$

Точка Q називається точкою кратності k , або просто – кратною точкою еліптичної кривої E , якщо для деякої точки P виконана рівність:

$$Q = P + \dots + P = kP \quad (4.10)$$

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розглянемо розроблене ПЗ керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure яке зображено на рисунку 5.1. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи:

- Навігаційне меню: Файл; Налаштування; Довідка.
- Розділу вкладок параметри.
- Вікно виведення результату роботи системи.
- Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші.
- Функціональних кнопок ПЗ.

Прикладом моделі політик може служити конфігурація міжмережного екрана, типова для більшості великих організацій: вона містить тисячі або навіть десятку тисяч рядків, і найчастіше про призначення багатьох з них уже ніхто не пам'ятає. На противагу цьому АСІ описує політику для додатків у термінах самих додатків і їхніх компонентів. Як ключові елементи для опису політики в інфраструктурі АСІ використовуються мережні профілі додатків (Application Network Profile), у яких задаються правила взаємодії між додатками або рівнями додатка. Останні представляються у вигляді груп елементів (End-Point Group, EPГ), що входять до складу додатка.

Політика, представлена в рамках мережного профілю додатка, описується в максимально абстрактних термінах – для цього не залучаються ні використовувані адреси, ні кількість елементів, що входять у групу, ні їхній тип (фізичні сервери, віртуальні машини або їхня комбінація). З одного боку, така політика може бути сформульована безпосередньо розроблювачами й адміністраторами додатків. Даний підхід відповідає широко обговорюваній в

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

останні роки моделі DevOps, відповідно до якої адміністратори додатків повинні взаємодіяти з інфраструктурою прямо, а не за допомогою серверних і мережних адміністраторів. З іншого боку, це дозволяє оперувати профілями додатків як логічними, абстрактними об'єктами, що допускають, скажемо, не тільки тиражування додатка різними замовниками при впровадженні декількох однотипних копій, але й копіювання між декількома ЦОД, незважаючи на використання різної адресації, розбіжність числа серверів і т.д.

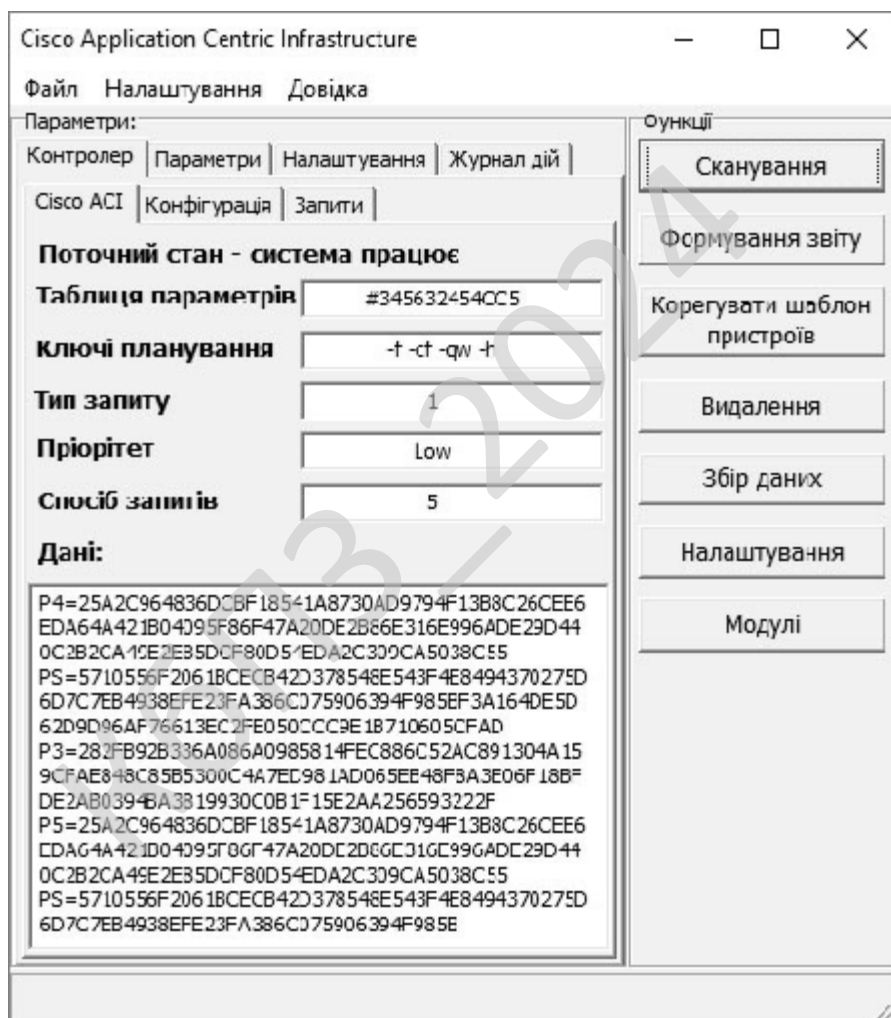


Рисунок 5.1 – Головне вікно ПЗ

Розроблена програма має дуже простий і інтуїтивно зрозумілий інтерфейс з користувачем. Кожен, хто в достатньому обсязі володіє операційним

середовищем Windows без особливих складностей освоїть і цю програму, оскільки її інтерфейс інтуїтивно зрозумілий.

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

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

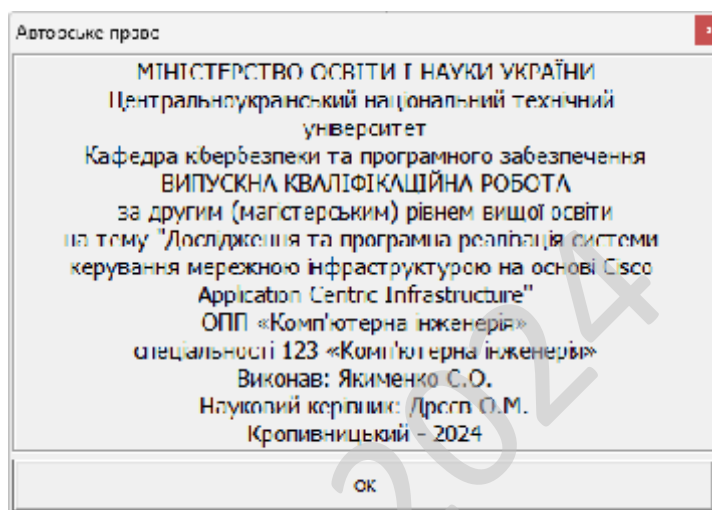


Рисунок 5.2 – Авторське право

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

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

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

вимог та характеристик. Розгортання може здійснюватись програмістом і в процесі розробки програмного забезпечення.

До діяльностей пов'язаних із розгортанням програмного забезпечення відносять:

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

При впровадженні програмного забезпечення потрібно урахувати наступні дії:

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

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

– Виконання робіт з загальної стандартизації існуючої діяльності організації, коли виділяються кращі практики виконання процедур і включаються в ІТ рішення за принципом найбільшої корисності для більшості учасників. Відсоток таких процедур щодо загального обсягу автоматизації може бути

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

невеликий, але це надає процесу побудови рішення вагу в організації за рахунок збільшення його необхідності.

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

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

Проводилась оцінка:

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

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

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

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

У цьому випадку формуються тестові варіанти, в яких:

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

Недоліки тестування "білої скриньки":

- Кількість незалежних маршрутів може бути дуже велика.

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

завантаження файлів через Інтернет або за носій (CD диск, флешку, ключ). Протягом певного терміну, що становить зазвичай тридцять днів, він може користуватися програмою, тестувати її, освоювати її можливості.

Якщо після закінчення цього терміну користувач вирішить продовжити використання ПЗ, він зобов'язаний купити його (zareestruvatisya), заплативши авторіві певну суму.

В іншому випадку користувач повинен припинити використання ПЗ та видалити його зі свого комп'ютера.

КБПЗ_2024

					VKPM-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure.

Метою розробки є дослідження та програмна реалізація системи керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure.

Об'єктом дослідження є процес керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure.

Предметом дослідження є методи керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure.

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

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

– Удосконалено метод керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure.

– Розроблено вітчизняний продукт керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

7 МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ІТ-ПРОЄКТУ

7.1 Визначення цільової аудиторії кінцевого готового продукту

Результати дослідження та програмна реалізація системи керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure можуть бути цікавими для таких категорій користувачів та організацій (рисуюнок 7.1).

Компанії та корпорації, що використовують приватні хмарні середовища — організації, які прагнуть оптимізувати та автоматизувати роботу своїх приватних хмарних рішень, зможуть зацікавитися новими підходами до інтеграції системи. Це може підвищити ефективність управління ресурсами та забезпечити кращу безпеку даних.

Розробники програмного забезпечення для хмарних платформ — ІТ-фахівці та розробники, що спеціалізуються на створенні, інтеграції та підтримці хмарних сервісів, можуть знайти корисними технічні інсайти та рішення, що оптимізують процеси взаємодії між різними компонентами хмари.

Інтегратори систем — компанії, які спеціалізуються на інтеграції технологічних рішень, можуть використовувати дане дослідження для підвищення якості своїх послуг, адаптуючи нові методики інтеграції та впровадження.

Організації з високими вимогами до безпеки — приватні хмарні рішення часто використовуються в секторах, де безпека є критично важливою, наприклад у банківській сфері, фінансових організаціях, медичних установах. Ці організації можуть скористатися новими рішеннями, що підвищують рівень захищеності даних.

Дослідники та академічна спільнота — аспіранти, студенти, науковці в галузі хмарних технологій, віртуалізації та безпеки можуть використовувати результати дослідження для подальших робіт та розробок.

Постачальники рішень для приватних хмар — компанії, що надають послуги хмарної інфраструктури на базі приватних хмар, можуть використовувати рішення для вдосконалення своїх пропозицій клієнтам.

Рисуюнок 7.1 – Цільова аудиторія

Цільова аудиторія залежить від специфіки реалізації і може бути розширена відповідно до потреб унікальних функцій та особливостей даного проекту.

7.2 Оцінка привабливості шляхом застосування методів експертних оцінок

Для оцінки привабливості програмної реалізації системи керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure можна застосувати метод експертних оцінок, який передбачає залучення фахівців для визначення ключових критеріїв і присвоєння їм вагових коефіцієнтів.

Вибираються найважливіші критерії, які будуть оцінювати експерти (рисунок 7.2).



Рисунок 7.2 – Критерії оцінки

Кожному критерію присвоюється ваговий коефіцієнт (за сумою ваг отримуємо 1 або 100%), що відображає його важливість для привабливості системи: функціональність (f): 0,3, масштабованість (s): 0,2, безпека (b): 0,25, економічна ефективність (e): 0,15, зручність використання (u): 0,1.

Кожен експерт (ІТ-архітектор, розробник, спеціаліст з безпеки) оцінює програмну реалізацію за шкалою від 1 до 10 (де 10 – найвищий бал) для кожного критерію. Зведені результати вносимо в таблицю 7.1.

Таблиця 7.1 – Зведені результати експертних оцінок

Критерій	Експерт 1	Експерт 2	Експерт 3	Середнє значення
Функціональність (F)	9	8	9	8,67
Масштабованість (S)	7	8	8	7,67
Безпека (B)	8	9	9	8,67
Економічна ефективність (E)	6	7	7	6,67
Зручність використання (U)	8	7	7	7,33

Для кожного критерію середнє значення оцінки множиться на ваговий коефіцієнт. Сума результатів дає загальну інтегральну оцінку привабливості системи.

$$\begin{aligned} \text{Загальна оцінка} &= (F \times 8,67) + (S \times 7,67) + (B \times 8,67) + (E \times 6,67) + (U \times 7,33) = \\ &= (0,3 \times 8,67) + (0,2 \times 7,67) + (0,25 \times 8,67) + (0,15 \times 6,67) + (0,1 \times 7,33) = 2,60 + 1,53 + 2,17 \\ &+ 1,00 + 0,73 = 8,03. \end{aligned}$$

Одержана інтегральна оцінка привабливості системи – 8,03 із 10 можливих, що свідчить про високу привабливість реалізації системи для інтеграції приватної хмари.

7.3 Вибір методу оцінки вартості ПЗ

Для оцінки вартості програмної реалізації системи керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure пропонуємо використовувати метод СОСОМО ІІ. Цей метод корисний для оцінки вартості великих проєктів, де враховуються обсяги коду, складність системи та специфіка команди. СОСОМО ІІ використовує емпіричні дані для прогнозування зусиль,

часу та витрат. Основні етапи: визначення обсягів роботи в кількості рядків коду, використання параметрів, таких як досвід команди, складність вимог, вимоги до надійності, розрахунок зусиль, часу і вартості, використовуючи формули СОСОМО II.

Для проекту з програмної реалізації керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure доцільно використовувати СОСОМО II адже саме він підходить для IT-проектів зі складною архітектурою та специфічними функціональними вимогами.

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

Впровадження системи керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure може призвести до значної економічної ефективності для організації. Варіанти економії подано на рисунку 7.3.

Впровадження системи інтеграції за рахунок сукупного зниження витрат і підвищення продуктивності принесе економію підприємству та підвищення ефективності роботи.

7.5 Пропозиція алгоритму просування проекту розробки ПЗ

Алгоритм просування проекту програмної реалізації системи керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure може включати кроки, представлені на рисунку 7.4.

Цей алгоритм дозволяє комплексно та послідовно реалізувати стратегію просування проекту системи керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure, забезпечуючи вихід на ринок і побудову стійких відносин з клієнтами та партнерами.

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

1. Аналіз ринку та визначення цільової аудиторії

- Дослідження ринку для визначення потреб компанії, що тів ориєнтовані приватні змири
- Визначення ключових характеристик цільової аудиторії: наприклад, кетаний і фінансовий, технічний, технологічний сферах, безпечна та інтеграція критично важливі
- Обсяг каналів і визначення основних каналів першого прототипу

2. Розробка унікальної торговельної пропозиції (УТП)

- Формулювання ключових переваг, які вирішують потреби цільових на ринку: наприклад, підвищення безпеки захищеність інтеграції економічних ресурсів
- Чітке визначення цінності пропозиції: платіжний, життєвий цикл інтеграції допоможе ми зрозуміти культури, швидкість адаптивності чи простоти адміністрування

3. Створення маркетингових матеріалів

- Розробка вебсайту чи іншого сторінки для презентації продукту з детальними елементами функцій, переваги та вигоди
- Створення брошури, презентації відеосюжетів, що демонструють функціональні переваги та переваги системи
- Підготовка технічних документів та інструкцій для спрощення впровадження та використання

4. Позичування продукту на виставках та конференціях

- Участь у професійних заходах, таких як IT-конференції, форуми з маркетингу, технології, щоб продемонструвати продукт цільовій аудиторії
- Проведення презентацій і живих демонстрацій для потенційних клієнтів та партнерів, щоб показати ефективність системи інтеграції

5. Заучення партнерів для розширення каналів збуту

- Пошук партнерів серед постачальників аналогічних рішень, інтеграторів, систем інтеграції, що співпрацюють з цільовою аудиторією, щоб розширити продажні результати
- Розробка партнерських програм і привабливих умовати співпраці, які сприятимуть впровадженню системи

6. Стратегія контент-маркетингу

- Підготовка технічних статей, оглядів об'єктів на платформах, які цікавлять цільову аудиторію, наприклад, Medium, LinkedIn, професійні журнали
- Створення кейс-стаді, case studies та бізнес-планів white papers, які демонструють успішні впровадження системи та її економічну ефективність
- Регулярне оновлення матеріалів для підтримки інтересу до продукту

7. Запуск рекламної кампанії

- Контекстна та таргетована реклама: Google Ads, LinkedIn та інші мережах для залучення цільових клієнтів
- Використання email-маркетингу для інформаційно: підтримки клієнтів, розміщення новин, оновлень та спеціальних пропозицій
- Рекламні для залучення аудиторії: живі відеозйомки сайт або онлайн-відеосюжетів з матеріалами презентації

8. Організація демонстраційних тестів (Proof of Concept)

- Надання потенційним клієнтам можливість провести тестові інтеграції PoC, щоб перевірити об'ємності та ефективності продукту в їхній інфраструктурі
- Підтримка на час тестування і допомога у вирішенні питань щодо налаштування тестів, використання

9. Збір відгуків та вдосконалення продукту

- Збір відгуків від перших клієнтів та партнерів, щоб зрозуміти потреби покращення
- Постійне вдосконалення продукту та адаптація до змін на ринку, щоб залишатися конкурентоспроможними

10. Вимірювання ефективності та коригування стратегії

- Аналіз показників, таких як конверсія на сайті, рівень задоволеності, соціальні мережах, діяльність нових партнерів, коштовість провадженню тощо
- Коригування стратегії просування на основі аналізу ефективності для досягнення кращих результатів

Рисунок 7.4 – Алгоритм просування проєкту

7.6 Оптимізація каналів збуту та шляхів реалізації ПЗ

Для оптимізації каналів збуту та шляхів реалізації проєкту програмної реалізації системи керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure можна запропонувати наступні стратегії (рисунок 7.4).

1. Аналіз та вибір ефективних каналів збуту

- **Прямий продаж через власну команду продажів:** Залучення фахівців з B2B-продажів для роботи з великими компаніями, що потребують інтеграції приватних хмарних систем. Прямий продаж особливо підходить для залучення великих корпоративних клієнтів, де потрібна детальна персоналізація рішення.
- **Партнерські канали:** Створення партнерських відносин із системними інтеграторами, провайдерами хмарних послуг та компаніями, що займаються безпекою. Це дозволить забезпечити вихід на ширшу аудиторію та збільшити обсяги збуту.
- **Інтернет-канали та електронна комерція:** Пропозиція ліцензій або демоверсій через сайт компанії, з можливістю оформлення заявки на тестування продукту. Це забезпечить залучення середніх і малих підприємств, які віддають перевагу самостійному ознайомленню з продуктом.

2. Оптимізація каналів збуту

- **Сегментація цільових клієнтів:** Виділення різних сегментів (великі корпорації, середні підприємства, державні установи) і адаптація підходів до кожного сегмента. Наприклад, для великих підприємств акцент на комплексні рішення та спеціалізовану технічну підтримку, а для малих компаній – готові до використання пакети з простою інтеграцією.
- **Розробка програми лояльності для партнерів:** Введення бонусів, знижок або пільгових умов для партнерів, що успішно реалізують продукт. Це може стимулювати більше партнерів активно просувати систему інтеграції.
- **Аутсорсинг продажів через зовнішні агентства:** Це може дозволити охопити ринки або сегменти, де компанія не має власних представництв. Агентства можуть займатися первинними продажами, а внутрішня команда – супроводом клієнтів та налаштуванням системи.

3. Оптимізація процесів залучення та утримання клієнтів

- **Надання демоверсій або безкоштовного тестування:** Можливість спробувати систему протягом обмеженого періоду може підвищити рівень довіри до продукту та допоможе клієнтам зрозуміти його цінність.
- **Регулярна комунікація та зворотний зв'язок:** Забезпечення чіткого каналу комунікації для обговорення функцій, вирішення проблем та адаптації до вимог клієнта, що дозволить утримувати клієнтів у довгостроковій перспективі.
- **Реферальні програми:** Запровадження програм, де клієнти можуть рекомендувати продукт іншим і отримувати бонуси або знижки. Це допоможе залучати нових клієнтів без значних витрат на маркетинг.

Рисунок 7.5 – Оптимізація каналів збуту та шляхів реалізації

Ці кроки допоможуть оптимізувати канали збуту та шляхи реалізації проекту програмної реалізації системи керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure, забезпечуючи максимальний охоплення цільової аудиторії, підвищення конкурентоспроможності продукту на ринку та зниження витрат на його просування.

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

7.7 Визначення ключових факторів успіху конкретного проєкту

Ключовими факторами успіху проєкту програмної реалізації системи керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure є ряд процедур та переваг (рисунок 7.6).



Рисунок 7.6 – Ключові фактори успіху проєкту

Ці фактори успіху дозволять проєкту не лише задовольнити потреби цільової аудиторії, але й виділитися на ринку як надійне, ефективне і вигідне рішення для керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure.

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Охорона праці в галузі ІТ-технологій дозволяє зберегти здоров'я, а іноді й життя працівників, тому є актуальною задачею.

Під час роботи з електронно-обчислювальними машинами (ЕОМ) на користувача у тій чи іншій мірі можуть впливати такі фізичні чинники:

- підвищені рівні змінного електромагнітного і електростатичного полів;
- підвищений рівень статичної електрики;
- підвищений рівень низько енергетичного (м'якого) рентгенівського іонізуючого випромінювання;
- підвищені рівні ультрафіолетового та інфрачервоного випромінювання;
- підвищений вміст позитивних аероіонів у повітрі робочої зони;
- знижений вміст негативних аероіонів;
- аномальний рівень освітленості робочої зони;
- підвищена яскравість фрагментів світлового зображення або світла, що потрапляє у поле зору користувача;
- підвищена нерівномірність розподілу яскравості у полі зору користувача;
- підвищена зовнішня освітленість екрану;
- підвищені пульсації світлового потоку джерел світла або світлового потоку, що випромінюється екраном;
- несприятливий для роботи спектр випромінювання джерел світла;
- підвищена часова нестабільність зображення; мерехтіння екрану;
- зміна яскравості свічення екрана;
- підвищена пряма блискіть, викликана попаданням у поле зору працюючого надмірно яскравого світла різних випромінюючих об'єктів;

- підвищена відбита блискість, обумовлена наявністю дзеркальних відбиття (відблисків), у тому числі від екрану;
- підвищений рівень шуму;
- аномальні температура, вологість та рухливість повітря робочої зони;
- підвищене значення напруги в електричному ланцюзі, замикання якого може статися через тіло людини;
- пожежа.

До недоліків умов праці користувачів комп'ютерної техніки можна віднести також недостатню площу і обсяг виробничого приміщення, недотримання вимог, мікроклімату на робочих місцях.

Праця користувачів ПЕОМ відносять до психічних форм праці з високим ступенем навантаження.

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

Ці шкідливі фактори можуть привести до професійних захворювань.

Відповідно до ст.14 Закону «Про охорони праці» [3] на роботодавця покладено обов'язок забезпечити:

- безпеку працівників при експлуатації устаткування;
- застосування коштів індивідуальної захисту працівників;
- відповідні вимоги охорони праці, умови праці в кожному робоче місце;
- дотримання режиму праці та відпочинку працівників;
- навчання безпечним методам і прийомам виконання;
- інструктажі з охорони праці;
- організацію контролю над станом умов праці в робочих місць;
- проведення атестації робочих місць в умовах праці.

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

Максимально зменшити кількість шкідливих впливів на людину при високій продуктивності праці, створити комфортні умови для роботи людей – одна з головних задач охорони праці [5].

8.2 Аналіз умов праці на робочому місці фахівця

Робота програміста пов'язана з постійною роботою на ЕОМ, яка відбувається у кімнаті розмірами 4,4 м×6,2 м×2,9 м. Одна з її більших стін має шість двостулкових вікон, розмірами 2 м×1,8 м, які виходять на північний захід. Вікна розташовані рівномірно по всій довжині стіни. Підлога в кімнаті покрита леноліумом, всі стіни пофарбовані світло оранжевого кольору до висоти 2,8 м, а далі підвісна стеля. Уздовж стін розташовані комп'ютерні столи. На них розташовуються 2 персональні комп'ютери й інша оргтехніка (сканер принтери, телефони й ксерокс). Столи мають пластикове покриття. Габарити їхньої робочої поверхні 1255 мм×845 мм. Висота столів 760 мм. Висота стільців від рівня підлоги становить 430 мм.

Згідно НПАОП 0.00 – 1.28 – 10 «Правила охорони праці під час електронно-обчислювальних машин» площа повинна задовольняти умові – не менш 6 м² на одне робоче місце. Кратність повітрообміну в приміщенні вузла також регламентується ДСанПіН 3.3.2.007-98 [2], вона повинна становити 20 м³/годину на одне місце. Виконання даних вимог забезпечить підтримку в приміщенні вузла оптимального значення вологості й складу повітря.

Відповідно ДБН В.2.5 – 28 – 2006 [1] роботу програміста можна віднести до роботи з малою точністю (найменший розмір об'єкта розрізнення від 1 до 5 мм) V-го розряду зорової роботи, з великою контрастністю об'єкта розрізнення (символів на екрані дисплея), з темним тлом (під розряд зорової роботи В). Приміщення вузла можна віднести до 1-ої групи приміщень, у яких проводиться розрізнення об'єктів зорової роботи при фіксованому напрямку лінії зору того, що працює на робочу поверхню. Для такого типу приміщень і розряду зорової роботи

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

нормоване значення коефіцієнта природної освітленості (КПО) робочої поверхні (при сполученому висвітленні), повинен становити 0,5%, освітленість при штучному висвітленні повинна становити 300 лк.

За результатами виміру освітленості відділом охорони праці величина освітленості від системи загального штучного висвітлення лежить у межах 200-250 лк, що не відповідає вимогам, які пред'являються до приміщення.

Відповідно ДСанПіН 3.3.2.007-98 [2] рівні звукового тиску в робочому приміщенні не повинні перевищувати в октавних смугах із середньогеометричними частотами наступних значень, наведених у таблиці 8.1.

Таблиця 8.1 – Допустимі спектри рівнів звукового тиску

Робоче місце	Рівень звукового тиску, дБ, в октавних смугах із середньогеометричними частотами, Гц								Рівень звуку і еквівалентний рівень звуку, дБА
	63	125	250	500	1000	2000	4000	8000	
Приміщення конструкторських бюро, програмістів обчислювальних машин, лабораторій для теоретичних робіт і опрацювання експериментальних даних, прийому хворих в медпунктах	71	61	54	49	45	42	40	38	50

У приміщенні перебувають наступні джерела шуму: електродвигуни внутрішнього вентилятора ЕОМ; працюючі принтери; працюючі дисководи. Шум, вироблений вентилятором можна класифікувати як постійний, всі інші джерела

Таблиця 8.3 – Параметри мікроклімату для приміщень, де встановлені комп'ютери

Період року	Параметр мікроклімату	Величина
Холодний	Температура повітря в приміщенні	22 – 24°C
	Відносна вологість	40 – 60%
	Швидкість руху повітря	до 0,1 м/с
Теплий	Температура повітря в приміщенні	23 – 25°C
	Відносна вологість	40 ... 60%
	Швидкість руху повітря	0,1 ... 0,2 м / с

8.3 Розробка заходів з умов поліпшення охорони праці

Провівши аналіз умов праці в розглянутому вище приміщенні, було отримано наступні результати:

- значення мікроклімату в приміщенні не перевищує норму;
- розрахунки розміру робочого місця на одного працівника відповідають нормі;
- рівень шуму в приміщенні не становить вище норми.

З вище перерахованих результатів можна зробити висновок, що основний вплив на продуктивність ІТ-фахівців є його психологічний стан. Тому є доцільним зменшити рівень стресу на робочому місці.

Рекомендовані наступні заходи: За потреби особливої концентрації уваги під час виконання робіт суміжні робочі місця операторів необхідно відділяти одне від одного перегородками висотою 1,5 – 2 м. Конструкція робочого місця користувача персонального комп'ютера має забезпечити підтримання оптимальної робочої пози офісного працівника. Конструкція робочого столу має відповідати сучасним вимогам ергономіки і забезпечувати оптимальне розміщення на робочій поверхні використовуваного обладнання (дисплея,

клавіатури, принтера) і документів. Висота робочої поверхні робочого столу має регулюватися в межах 680-800 мм, а ширина і глибина – забезпечувати можливість виконання операцій у зоні досяжності моторного поля (рекомендовані розміри: 600-1400мм, глибина – 800-1000мм). Робочий стіл повинен мати простір для ніг заввишки не менше ніж 600 мм, завширшки не менше ніж 500 мм, завглибшки (на рівні колін) не менше ніж 450 мм, на рівні простягнутої ноги не менше ніж 650 мм. Робочий стілець має бути підйомно-поворотним, регульованим за висотою, з кутом і нахилу сидіння та спинки і за відстанню від спинки до переднього краю сидіння поверхня сидіння має бути плоскою, передній край – заокругленим [7].

8.4 Розрахункова частина

Для захисного штучного заземлення застосовуються вертикальні електроди: металевий куток 80x80x8 мм, (згідно з ДСТУ 2251:2018 «Кутики сталеві гарячекатані рівнополічні. Сортамент») довжиною $L=1,6$ м та горизонтальний електрод – металева полоса з перетином 60x5 мм. Напруга – 220/380 В. Розрахункова схема розташування заземлюючих електродів – у ряд (рис. 8.1).

Розрахунок проведемо за допустимим опором розтіканню струму заземлювача.

Початкові дані для розрахунку захисного заземлення: тип верхнього шару ґрунту – чорнозем, нижнього шару ґрунту – глина (питомий опір $\rho_2 = 40$ Ом·м). Умовна товщина верхнього шару ґрунту: $H=0,5$ м. Відстань між вертикальними заземлювачами (електродами) $A=1,6$ м. Глибина закладення горизонтального контура заземлення $t=0,6$ м. Опір заземлювача, який нормується: $R_{3H} = 4$ Ом. Необхідно визначити необхідну кількість вертикальних заземлювачів та довжину полоси (горизонтального заземлювача).

Виконаємо розрахунок.

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

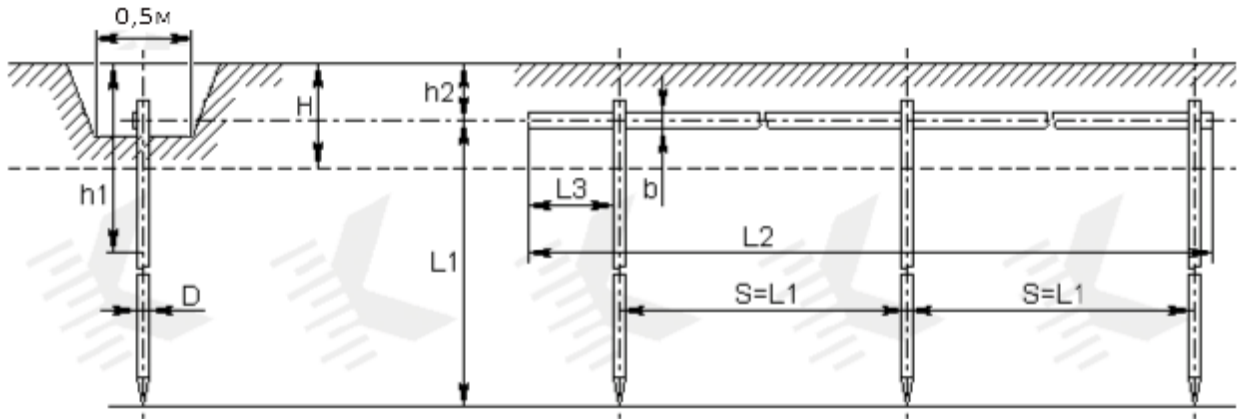


Рисунок 8.1 – Схема штучного заземлення.

Відстань від центра вертикального заземлювача до поверхні землі:

$$T = t + L/2 = 0,6 + 1,6/2 = 1,4 \text{ м.}$$

Розрахунковий питомий опір ґрунту (з врахуванням того, що фактично вся конструкція заземлювача розташовується у нижньому шарі ґрунту):

$$\rho = \psi \cdot \rho = 1,36 \cdot 40 = 54,5 \text{ Ом} \cdot \text{м.}$$

де $\psi = 1,36$ – табличне значення коефіцієнта сезонності для відповідної кліматичної зони у багатошаровому ґрунті [6];

$\rho_2 = 40 \text{ Ом} \cdot \text{м.}$ – табличне значення питомого опору нижнього шару ґрунту (глина) [7].

Еквівалентний діаметр вертикального електрода (кутка) [7]:

$$D_{\text{в}} = 0,95 \cdot K = 0,95 \cdot 80 = 76,85 \text{ мм} = 0,076 \text{ м.}$$

де $K = 80 \text{ мм}$ – розмір полиці металевого кутка.

Відношення $A/L = 1,6/1,6 = 1$.

Опір розтіканню електричного струму одного електрода вертикального заземлювача з урахуванням заглиблення заземлювача [7]:

$$R_0 = 0,366 \cdot (\rho/L) \cdot [\lg(2L/D_{\text{в}}) + (1/2) \cdot \lg((4T+L)/(4T-L))] = \\ = 0,366 \cdot (54,5/1,6) \cdot [\lg(2 \cdot 1,6/0,076) + (1/2) \cdot \lg((4 \cdot 1,4+1,6)/(4 \cdot 1,4-1,6))] = 21,8 \text{ Ом.}$$

Визначаємо коефіцієнт екранування вертикальних електродів $K_{\text{ев}} = 0,8$ при орієнтовній кількості вертикальних електродів, яка дорівнює 5 [7].

Визначаємо необхідну кількість вертикальних електродів заземлювача (без врахування горизонтального заземлювача), при $R_{\text{ЗН}} = 4 \text{ Ом}$:

$$N=R_0 / (K_{ев} \cdot R_{3н})= 21,8 / (0,8 \cdot 4) = 6,8 \approx 7 \text{ шт.}$$

Визначаємо довжину з'єднуючої полоси:

$$L_{п} = 1,05 \cdot A \cdot N = 1,05 \cdot 1,6 \cdot 7 = 12,2 \approx 12 \text{ м.}$$

Опір розтіканню електричного струму з'єднуючої полоси з урахуванням кліматичного коефіцієнту питомого опору ґрунту $K_{п}$ [7]:

$$\begin{aligned} R_{п} &= 0,366 \cdot (\rho \cdot K_{п} / L_{п}) \cdot \lg(2 \cdot L_{п}^2 / (B \cdot t)) = \\ &= 0,366 \cdot (40 \cdot 5 / 12) \cdot \lg((2 \cdot 12^2) / (0,06 \cdot 0,6)) = 20,16 \text{ Ом.} \end{aligned}$$

де $K_{п}=5$ – табличне значення кліматичного коефіцієнту питомого опору ґрунту для відповідної кліматичної зони для з'єднуючої полоси [7]:

$B = 60 \text{ мм.} = 0,06 \text{ м}$ – ширина з'єднуючої полоси (задана).

Загальний опір розтіканню електричного струму заземлювача [7]:

$$\begin{aligned} R &= (R_0 \cdot R_{п}) / (R_0 \cdot \eta_{п} + N \cdot R_{п} \cdot K_{ев}) = \\ &= (21,8 \cdot 20,16) / (21,8 \cdot 0,75 + 9 \cdot 20,16 \cdot 0,8) = 3,46 \text{ Ом.} \end{aligned}$$

де $\eta_{п} = 0,75$ – табличне значення коефіцієнта екранування з'єднуючої полоси [7].

Умова $R \leq R_{3н}$ виконується ($3,46 \leq 4$).

Оскільки при 7 вертикальних електродах R суттєво менше $R_{3н}$, зменшимо кількість вертикальних електродів N до 6 і виконаємо перерахунок. У результаті остаточно отримали: $R = 3,96 \text{ Ом.}$ при кількості вертикальних електродів $N=6$.

8.5 Висновки до розділу

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

З цих міркувань було здійснено аналіз умов праці на робочому місці програміста, проведено розгляд небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи.

Виконано розрахунок захисного штучного заземлення, як одного з ключових факторів безпеки програміста.

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure.

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

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти наведені теоретичне узагальнення й рішення наукового завдання дослідження методів керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure.
- Досліджена система керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure.
- На основі отриманих результатів досліджень створена програмна реалізація системи керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure.

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

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

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

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

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм ДСТУ 4145-2002.

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

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

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		86

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Якименко С.О. Дослідження та програмна реалізація системи керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2024.
2. Оліфер В.Г. Комп'ютерні мережі. Принципи, технології, протоколи. Підручник / В.Г. Оліфер, Н.А.Оліфер. – [5-е вид.]. – 2016. – 944 с.
3. Е. Таненбаум, Д. Уезеролл «Комп'ютерні мережі». – [5-е вид.]. – 2016. – 960 с.
4. Wendell Odom. «CCNA 200-301 Official Cert Guide, Volume 1». Cisco Press. 2020. – 848 p.
5. Wendell Odom. «CCNA 200-301 Official Cert Guide, Volume 2 Premium Edition eBook and Practice Test». Cisco Press. 2020. – 624 p.
6. Scott Jernigan «CompTIA Network+ Certification All-in-One Exam Guide, Eighth Edition». 2022. – 976 p.
7. Doug Lowe «Networking For Dummies 12th Edition». 2020. – 480 p.
8. Ramon Nastase «Computer Networking: The Beginner's guide for Mastering Computer Networking, the Internet and the OSI Model». 2018. – 186 p.
9. Russ White & Ethan Banks «Computer Networking Problems and Solutions: An Innovative Approach to Building Resilient, Modern Networks». 2017. – 832 p.
10. 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.
11. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		87

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

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

13. 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.

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. 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.

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

17. 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.

18. 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.

19. 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.

20. 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.

21. 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.

22. 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.

23. 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.

24. 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.

25. 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.

26. 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.

27. 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.

28. 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.

29. 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.

30. 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.

31. 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.

32. 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.

33. 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.

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

35. 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.

36. 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.

37. 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.

38. 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.

39. 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.

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

40. 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

41. 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.

42. 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.

43. 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.

44. 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.

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

46. 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.

47. 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

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

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).

48. 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

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

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

51. 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.

					ВКРМ-123.24.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

Додаток А
(обов'язковий)

Технічне завдання

Зміст

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

					ВКРМ-123.24.0050.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Якименко С.О.				<i>Дослідження та програмна реалізація системи керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure</i>	Літ.	Аркуш	Аркушів
Перевірів	Дресв О.М.					М	1	6
Н. Контр.	Коваленко А.С.					ЦНТУ КІ-23М		
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure.

2 Підстава для розробки

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

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure.

4 Джерела розробки

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

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

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

					ВКРМ-123.24.0050.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

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

5.2 Показники призначення

Система повинна забезпечувати:

- програмну реалізацію системи керування мережною інфраструктурою на основі Cisco Application Centric Infrastructure;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

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

5.4 Вимоги до архітектури

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

5.5 Вимоги до надійності

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

					ВКРМ-123.24.0050.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

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

5.7 Вимоги до складу та параметрів технічних засобів

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

5.8 Вимоги до інформаційної і програмної сумісності

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

5.8.1 Обладнання

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

5.8.2 Мова програмування

Середовище Builder C++.

					ВКРМ-123.24.0050.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

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

7 Економічні вимоги

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

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

8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинен бути розглянутий аналіз умов праці на робочому місці фахівця.

					ВКРМ-123.24.0050.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

9 Перелік документів, що розробляються

– Наукова новизна	– 1 аркуш.
– Структурна схема системи	– 1 аркуш.
– Функціональна схема системи	– 1 аркуш.
– Діаграма процесів	– 1 аркуш.
– Блок-схема алгоритму роботи програми	– 2 аркуша.
– Показники економічної ефективності	– 1 аркуш.
– Пояснювальна записка	– 93 аркуші.

10 Етапи розробки

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

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

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

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Маркетингове та економічне обґрунтування ІТ-проєкту.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

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

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 19.12.2024 р.

					ВКРМ-123.24.0050.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

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

_____ Дреєв О.М.

*Дослідження та програмна реалізація
системи керування мережною інфраструктурою на основі Cisco Application
Centric Infrastructure*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 68

Літера: РП

Кропивницький – 2024 року

resalloc.cpp - Розподіл ресурсів у системі

```

// системний файл: resalloc.cpp

#include "errno.h"

#include "tsys.h"
#include "resalloc.h"

using namespace Cisco_Application_Centric_Infrastructure;

//*****
//* Об'єкт ресурсу *
//*****
Res::Res( )
{
    #if !__GLIBC_PREREQ(2,4)
        wThr = 0;
    #endif
    if(pthread_rwlock_init(&rw, NULL))
        throw TError("ResAlloc", _("Помилка відкриття семафору!"));
}

Res::~Res( )
{
    pthread_rwlock_wrlock(&rw);
    pthread_rwlock_destroy(&rw);
}

void Res::resRequestW( unsigned short tm )
{
    int rez = 0;
    #if !__GLIBC_PREREQ(2,4)
        //EDEADLK імітація
        if(wThr && wThr == pthread_self()) rez == EDEADLK;
    else
    #endif
        if(!tm) rez = pthread_rwlock_wrlock(&rw);
    else
    {
        timespec wtm;
        clock_gettime(CLOCK_REALTIME, &wtm);
        wtm.tv_nsec += 1000000*(tm%1000);
        wtm.tv_sec += tm/1000 + wtm.tv_nsec/1000000000; wtm.tv_nsec =
        wtm.tv_nsec%1000000000;
        rez = pthread_rwlock_timedwrlock(&rw, &wtm);
    }
    if(rez == EDEADLK) throw TError(10, "ResAlloc", _("Ресурс пробує заблокувати потік!"));
    else if(tm && rez == ETIMEDOUT) throw TError("ResAlloc", _("Ресурс не відповідає!"));
    #if !__GLIBC_PREREQ(2,4)
        wThr = pthread_self();
    #endif
}

bool Res::resTryW( )
{
    int rez = pthread_rwlock_trywrlock(&rw);
    if(rez == EBUSY) return false;
    else if(rez == EDEADLK) throw TError(10, "ResAlloc", _("Ресурс пробує заблокувати потік!"));
    return true;
}

```

```

void Res::resRequestR( unsigned short tm )
{
    int rez = 0;
#ifdef __GLIBC__
    #if !__GLIBC__PREREQ(2,4)
        //EDEADLK імітація
        if(wThr && wThr == pthread_self()) rez == EDEADLK;
    else
    #endif
    if(!tm) rez = pthread_rwlock_rdlock(&rw);
    else
    {
        timespec wtm;
        clock_gettime(CLOCK_REALTIME, &wtm);
        wtm.tv_nsec += 1000000*(tm%1000);
        wtm.tv_sec += tm/1000 + wtm.tv_nsec/1000000000; wtm.tv_nsec =
wtm.tv_nsec%1000000000;
        rez = pthread_rwlock_timedrdlock(&rw, &wtm);
    }
    if(rez == EDEADLK) throw TError(10, "ResAlloc", _("Ресурс пробує заблокувати
потік!"));
    else if(tm && rez == ETIMEDOUT) throw TError("ResAlloc", _("Ресурс не
відповідає!"));
}

bool Res::resTryR( )
{
    int rez = pthread_rwlock_tryrdlock(&rw);
    if(rez == EBUSY) return false;
    else if(rez == EDEADLK) throw TError(10, "ResAlloc", _("Ресурс пробує
заблокувати потік!"));
    return true;
}

void Res::resRelease( )
{
    pthread_rwlock_unlock(&rw);
#ifdef __GLIBC__
    #if !__GLIBC__PREREQ(2,4)
        if(wThr == pthread_self()) wThr = 0;
    #endif
}

//*****
//* Автоматичний розподіл ресурсів *
//*****
ResAlloc::ResAlloc( Res &rid ) : mId(rid), mAlloc(false)
{
}

ResAlloc::ResAlloc( Res &rid, bool write, unsigned short tm ) : mId(rid),
mAlloc(false)
{
    request(write, tm);
}

ResAlloc::~ResAlloc( )
{
    if(mAlloc) release();
}

void ResAlloc::request( bool write, unsigned short tm )
{
    if(mAlloc) release();
    mAlloc = false;
    try
    {
        if(write) mId.resRequestW(tm);
        else mId.resRequestR(tm);
        mAlloc = true;
    }
}

```

```

        }catch(TError err) { if(err.cod!=10) throw; }
    }

void ResAlloc::release()
{
    if(!mAlloc) return;
    mId.resRelease( );
    mAlloc = false;
}

//*****
/* Рядок + ресурс для *
//*****
ResString::ResString( const string &vl )
{
    pthread_mutex_init(&mRes, NULL);
    setVal(vl);
}

ResString::~ResString( )
{
    pthread_mutex_lock(&mRes);
    pthread_mutex_destroy(&mRes);
}

size_t ResString::size( )      { return getVal().size(); }

bool ResString::empty( )      { return getVal().empty(); }

void ResString::setVal( const string &vl )
{
    pthread_mutex_lock(&mRes);
    str = vl;
    pthread_mutex_unlock(&mRes);
}

string ResString::getVal( )
{
    string rez;
    pthread_mutex_lock(&mRes);
    rez = str;
    pthread_mutex_unlock(&mRes);
    return rez;
}

ResString &ResString::operator=( const string &val )
{
    setVal(val);
    return *this;
}

```

tarchives.cpp - Робота з архівом

```

//WebCisco_Application_Centric_Infrastructure системний файл: tarchives.cpp

#include <unistd.h>
#include <getopt.h>
#include <signal.h>
#include <sys/time.h>
#include <string.h>
#include <algorithm>

#include "tsys.h"
#include "tarchives.h"

#define BUF_SIZE_DEF 500
#define BUF_SIZE_MAX 100000

using namespace Cisco_Application_Centric_Infrastructure;

//*****
//* Підсистема архівування *
//*****

//*****
//* TArchiveS *
//*****
TArchiveS::TArchiveS( ) :
    TSubSYS(SARH_ID,"Archives",true), elMess(""), elVal(""), elAval(""),
    bufErr(0), mMessPer(10), prcStMess(false),
    headBuf(0), headLstread(0), mValPer(1000), mValPrior(10), prcStVal(false),
    endrunReqVal(false)
{
    mAval = grpAdd("va_");

    //> архіватор повідомлення у структурі БД
    elMess.fldAdd( new TFld("ID",_("ID"),TFld::String,TCfg::Key,"20") );
    elMess.fldAdd( new TFld("MODUL",_("Ім'я модуля
(плагіна)"),TFld::String,TCfg::Key,"20") );
    elMess.fldAdd( new
TFld("NAME",_("Ім'я"),TFld::String,TCfg::TransltText,"50") );
    elMess.fldAdd( new
TFld("DESCR",_("Дескриптор"),TFld::String,TCfg::TransltText,"200") );
    elMess.fldAdd( new TFld("START",_("Початок архіву"),TFld::Boolean,0,"1") );
    elMess.fldAdd( new TFld("CATEG",_("Категорії
повідомлень"),TFld::String,0,"100") );
    elMess.fldAdd( new TFld("LEVEL",_("Рівні
повідомлень"),TFld::Integer,0,"1","","0;7") );
    elMess.fldAdd( new TFld("ADDR",_("Адреса"),TFld::String,0,"100") );

    //> Значення архіватора у структурі БД
    elVal.fldAdd( new TFld("ID",_("ID"),TFld::String,TCfg::Key,"20") );
    elVal.fldAdd( new TFld("MODUL",_("Ім'я модуля
(плагіна)"),TFld::String,TCfg::Key,"20") );
    elVal.fldAdd( new TFld("NAME",_("Ім'я
"),TFld::String,TCfg::TransltText,"50") );
    elVal.fldAdd( new
TFld("DESCR",_("Дескриптор"),TFld::String,TCfg::TransltText,"200") );
    elVal.fldAdd( new TFld("START",_("Початок архіву"),TFld::Boolean,0,"1","0")
);
    elVal.fldAdd( new TFld("ADDR",_("Адреса"),TFld::String,0,"50") );
    elVal.fldAdd( new TFld("V_PER",_("Value period
(sec)"),TFld::Real,0,"12.6","1","0;1000000") );
    elVal.fldAdd( new TFld("A_PER",_("Період архівації
(sec)"),TFld::Integer,0,"4","60","0;1000") );

    //> Значення архіву у структурі БД
    elAval.fldAdd( new TFld("ID",_("ID"),TFld::String,TCfg::Key,"20") );

```

```

    elAval.fldAdd( new TFld("NAME",_(" Ім'я
"),TFld::String,TCfg::TransltText,"50") );
    elAval.fldAdd( new
TFld("DESCR",_("Дескриптор"),TFld::String,TCfg::TransltText,"200") );
    elAval.fldAdd( new TFld("START",_("Початок архіву"),TFld::Boolean,0,"1","0")
);
    elAval.fldAdd( new TFld("SrcMode",_("Режим джерела"),TFld::Integer,0,"1") );
    elAval.fldAdd( new TFld("Source",_("Джерело"),TFld::String,0,"100") );
    elAval.fldAdd( new TFld("VTYPE",_("Тип значення"),TFld::Integer,0,"1") );
    elAval.fldAdd( new TFld("BPER",_("Період буферізації
(sec)"),TFld::Real,0,"9.6","1","0;10000") );
    elAval.fldAdd( new TFld("BSIZE",_("Розмір
буферу()"),TFld::Integer,0,"6","100","0;1000000") );
    elAval.fldAdd( new TFld("BHGRD",_("Буфер у режимі ґрид-
системи"),TFld::Boolean,0,"1","1") );
    elAval.fldAdd( new TFld("BHRES",_("Значення буфера останім
часом"),TFld::Boolean,0,"1","0") );
    elAval.fldAdd( new TFld("ArchS",_("Процес
архівування"),TFld::String,0,"500") );

    setMessBufLen( BUF_SIZE_DEF );

    //> Створення повідомлення часу архівування
    struct sigevent sigev;
    memset(&sigev,0,sizeof(sigev));
    sigev.sigev_notify = SIGEV_THREAD;
    sigev.sigev_value.sival_ptr = this;
    sigev.sigev_notify_function = ArhMessTask;
    sigev.sigev_notify_attributes = NULL;
    timer_create(CLOCK_REALTIME,&sigev,&tmIdMess);
}

TArchiveS::~TArchiveS( )
{
    //> Повідомлення про закінчення архівування
    timer_delete(tmIdMess);

    //> Переривання архівування
    if(prcStVal) SYS->taskDestroy(nodePath('.',true)+".vals", &endrunReqVal);

    //> Визволення усіх ресурсів
    nodeDelAll();
}

int TArchiveS::valPeriod( )          { return vmax(1,mValPer); }

void TArchiveS::setValPrior( int ivl ) { mValPrior = vmax(-1,vmin(99,ivl));
modif(); }

void TArchiveS::load_( )
{
    //> Завантажуємо параметри з командного рядка
    int next_opt;
    const char *short_opt="h";
    struct option long_opt[] =
    {
        {"help"      ,0,NULL,'h'},
        {NULL        ,0,NULL,0 }
    };

    optind=0,opterr=0;
    do
    {
        next_opt=getopt_long(SYS->argc,(char * const *)SYS-
>argv,short_opt,long_opt,NULL);
        switch(next_opt)
        {
            case 'h': fprintf(stdout,"%s",optDescr().c_str()); break;
            case -1 : break;

```

```

    }
    } while(next_opt != -1);

    //> Завантажуємо параметри
    setMessBufLen (
atoi (TBDS::genDBGet (nodePath ()+"MessBufSize", TSYs::int2str (messBufLen ())) .c_str (
)) );
    setMessPeriod (
atoi (TBDS::genDBGet (nodePath ()+"MessPeriod", TSYs::int2str (mMessPer)) .c_str ()) );
    setValPeriod (
atoi (TBDS::genDBGet (nodePath ()+"ValPeriod", TSYs::int2str (mValPer)) .c_str ()) );
    setValPrior (
atoi (TBDS::genDBGet (nodePath ()+"ValPriority", TSYs::int2str (mValPrior)) .c_str ())
);

    //> LidDB
    //>> Повідомлення завантаження архіватора
    string id,type;
    map<string, bool> itReg;
    try
    {
        TConfig c_el (&elMess);
        c_el.cfgViewAll (false);
        vector<string> db_ls;

        //>> Шукаємо у БД і створюємо новий архів
        SYS->db ().at ().dbList (db_ls, true);
        db_ls.push_back ("");
        for (unsigned i_db = 0; i_db < db_ls.size (); i_db++)
            for (int fld_cnt=0; SYS-
>db ().at ().dataSeek (db_ls [i_db] + "." + subId () + "_mess_proc", nodePath () + subId () + "_me
ss_proc", fld_cnt++, c_el); )
            {
                id = c_el.cfg ("ID").getS ();
                type = c_el.cfg ("MODUL").getS ();
                if (modPresent (type) && !at (type).at ().messPresent (id))
                    at (type).at ().messAdd (id, (db_ls [i_db] == SYS-
>workDB ()) ? "*" : db_ls [i_db]);
                itReg [type + "." + id] = true;
            }

        //>>> Перевіряємо для видалення з БД
        if (!SYS->selDB ().empty ())
        {
            vector<string> m_ls;
            modList (m_ls);
            for (unsigned i_m = 0; i_m < m_ls.size (); i_m++)
            {
                at (m_ls [i_m]).at ().messList (db_ls);
                for (unsigned i_it = 0; i_it < db_ls.size (); i_it++)
                    if (itReg.find (m_ls [i_m] + "." + db_ls [i_it]) == itReg.end () &&
SYS->chkSelDB (at (m_ls [i_m]).at ().messAt (db_ls [i_it]).at ().DB ()))
                        at (m_ls [i_m]).at ().messDel (db_ls [i_it]);
            }
        }
    } catch ( TError err )
    {
        mess_err (err.cat.c_str (), "%s", err.mess.c_str ());
        mess_err (nodePath ().c_str (), _ ("Повідомлення вомилки завантаження
архіватора."));
    }

    //>> Завантажуємо значення архіватора
    try
    {
        TConfig c_el (&elVal);
        c_el.cfgViewAll (false);
        vector<string> db_ls;
        itReg.clear ();
    }

```

```

//>> Шукаємо у БД та створюємо новий архів
SYS->db().at().dbList(db_ls,true);
db_ls.push_back("<cfg>");
for(unsigned i_db = 0; i_db < db_ls.size(); i_db++)
    for(int fld_cnt=0; SYS-
>db().at().dataSeek(db_ls[i_db]+"."+subId()+"_val_proc",nodePath()+subId()+"_val
_proc",fld_cnt++,c_el); )
    {
        id = c_el.cfg("ID").getS();
        type = c_el.cfg("MODUL").getS();
        if(modPresent(type) && !at(type).at().valPresent(id))
            at(type).at().valAdd(id,(db_ls[i_db]==SYS-
>workDB())?"*.*":db_ls[i_db]);
        itReg[type+"."+id] = true;
    }

//>>> Перевіряємо для видалення з БД
if(!SYS->selDB().empty())
    {
        vector<string> m_ls;
        modList(m_ls);
        for(unsigned i_m = 0; i_m < m_ls.size(); i_m++)
            {
                at(m_ls[i_m]).at().valList(db_ls);
                for(unsigned i_it = 0; i_it < db_ls.size(); i_it++)
                    if(itReg.find(m_ls[i_m]+"."+db_ls[i_it]) == itReg.end() &&
SYS->chkSelDB(at(m_ls[i_m]).at().valAt(db_ls[i_it]).at().DB()))
                        at(m_ls[i_m]).at().valDel(db_ls[i_it]);
            }
    }
} catch( TError err )
{
    mess_err(err.cat.c_str(),"%s",err.mess.c_str());
    mess_err(nodePath().c_str(),"Помилка завантаження значення
архіватора.");
}

//>> Завантажуємо значення архіватора
try
{
    TConfig c_el(&selAval);
    c_el.cfgViewAll(false);
    vector<string> db_ls;
    itReg.clear();

    //>> Шукаємо у БД та створюємо новий архів
SYS->db().at().dbList(db_ls,true);
db_ls.push_back("<cfg>");
for(unsigned i_db = 0; i_db < db_ls.size(); i_db++)
    for(int fld_cnt=0; SYS-
>db().at().dataSeek(db_ls[i_db]+"."+subId()+"_val",nodePath()+subId()+"_val",fld
_cnt++,c_el); )
    {
        id = c_el.cfg("ID").getS();
        if(!valPresent(id)) valAdd(id,(db_ls[i_db]==SYS-
>workDB())?"*.*":db_ls[i_db]);
        itReg[id] = true;
    }

    //>>> Перевіряємо для видалення з БД
if(!SYS->selDB().empty())
    {
        valList(db_ls);
        for(unsigned i_it = 0; i_it < db_ls.size(); i_it++)
            if(itReg.find(db_ls[i_it]) == itReg.end() && SYS-
>chkSelDB(valAt(db_ls[i_it]).at().DB()))
                valDel(db_ls[i_it]);
    }
}

```

```

    }catch(TError err)
    {
        mess_err(err.cat.c_str(),"%s",err.mess.c_str());
        mess_err(nodePath().c_str(),_("Помилка завантаження значення
архіватора."));
    }
}

void TArchiveS::save_( )
{
    vector<string> t_lst, o_lst;

    //> Зберігаємо параметри
    TBDS::genDBSet (nodePath()+"MessBufSize",TSYS::int2str(messBufLen()));
    TBDS::genDBSet (nodePath()+"MessPeriod",TSYS::int2str(messPeriod()));
    TBDS::genDBSet (nodePath()+"ValPeriod",TSYS::int2str(valPeriod()));
    TBDS::genDBSet (nodePath()+"ValPriority",TSYS::int2str(valPrior()));
}

void TArchiveS::valAdd( const string &iid, const string &idb )
{
    if( valPresent(iid) ) return;
    chldAdd(mAval,new TVArchive(iid,idb,&aVale()));
}

string TArchiveS::optDescr( )
{
    char buf[STR_BUF_LEN];
    snprintf(buf,sizeof(buf),_(
        "===== Підсистема \"Архів\" Опції
=====\\n"
        "----- Параметри частини '%s' у конфігураційний файл -----\\n"
        "MessBufSize <items>      Повідомлення розміру буферу.\\n"
        "MessPeriod <sec>          Повідомлення періоду архівування.\\n"
        "ValPeriod <msec>          Значення періоду архівування.\\n"
        "ValPriority <level>       Значення завдання пріоритетного рівня.\\n"
        "MaxReqMess <items>       Повідомлення максимального запиту.\\n"
        "MaxReqVals <items>       Значення максимального запиту.\\n\\n"
    ),nodePath().c_str());

    return buf;
}

void TArchiveS::subStart( )
{
    mess_info(nodePath().c_str(),_("Запускаємо підсистеми."));

    SubStarting = true;

    vector<string> t_lst, o_lst;

    modList(t_lst);
    for(unsigned i_t = 0; i_t < t_lst.size(); i_t++)
    {
        AutoHD<TTipArchivator> mod = modAt(t_lst[i_t]);

        //> Повідомлення про початок роботи архіватора
        mod.at().messList(o_lst);
        for(unsigned i_o = 0; i_o < o_lst.size(); i_o++)
        {
            AutoHD<TMArchivator> mess = mod.at().messAt(o_lst[i_o]);
            if( /*!mess.at().startStat() &&*/ mess.at().toStart() )
                try{ mess.at().start(); }
                catch(TError err)
                {
                    mess_err(err.cat.c_str(),"%s",err.mess.c_str());
                    mess_err(nodePath().c_str(),_("Повідомлення про помилку роботи
архіватора."),o_lst[i_o].c_str());
                }
        }
    }
}

```

```

}
//> Значення початку роботи архіватора
mod.at().valList(o_lst);
for(unsigned i_o = 0; i_o < o_lst.size(); i_o++)
{
    AutoHD<TVArchivator> val = mod.at().valAt(o_lst[i_o]);
    if( /*!val.at().startStat() &&*/ val.at().toStart() )
        try{ val.at().start(); }
        catch(TError err)
        {
            mess_err(err.cat.c_str(), "%s", err.mess.c_str());
            mess_err(nodePath().c_str(), _("Помилка початку роботи
архіватора."), val.at().workId().c_str());
        }
    }
}

//> Значення початку роботи архіватора
vallList(o_lst);
for(unsigned i_o = 0; i_o < o_lst.size(); i_o++)
{
    AutoHD<TVArchive> aval = valAt(o_lst[i_o]);
    if( /*!aval.at().startStat() &&*/ aval.at().toStart() )
        try{ aval.at().start(); }
        catch(TError err)
        {
            mess_err(err.cat.c_str(), "%s", err.mess.c_str());
            mess_err(nodePath().c_str(), _("Помилка початку роботи
архіватора"), o_lst[i_o].c_str());
        }
    }

//> Повідомлення про початок роботи інтервального таймера
struct itimerspec itval;
itval.it_interval.tv_sec = itval.it_value.tv_sec = messPeriod();
itval.it_interval.tv_nsec = itval.it_value.tv_nsec = 0;
timer_settime(tmIdMess, 0, &itval, NULL);

//> Значення початку роботи завдань
if(!prcStVal) SYS->taskCreate(nodePath('.', true) + ".vals", valPrior(),
TArchiveS::ArhValTask, this);

TSubSYS::subStart( );

SubStarting = false;
}

void TArchiveS::subStop( )
{
    mess_info(nodePath().c_str(), _("Зупинка підсистеми.));

    TSubSYS::subStop( );

    vector<string> t_lst, o_lst;

    //> Зупинка інтервального таймера для періодичних потоків, для створення
повідомлення архівації структур;
    itval.it_interval.tv_sec = itval.it_value.tv_sec =
        itval.it_interval.tv_nsec = itval.it_value.tv_nsec = 0;
    timer_settime(tmIdMess, 0, &itval, NULL);
    if(TSYS::eventWait( prcStMess, false, nodePath()+"mess_stop", 10))
        throw TError(nodePath().c_str(), _("Архівація повідомлень потоків не може
бути зупинена!"));

    //> Значення зупинки роботи завдань
    if(prcStVal) SYS->taskDestroy(nodePath('.', true) + ".vals", &endrunReqVal);

    //> Виклик останнього повідомлення архіватора
    sigval obj; obj.sival_ptr = this;

```

```

ArhMessTask(obj);

//> Зупинка архіватора
modList(t_lst);
for(unsigned i_t = 0; i_t < t_lst.size(); i_t++)
{
    AutoHD<TTipArchivator> mod = modAt(t_lst[i_t]);
    //Значення зупинки архіватора
    mod.at().vallList(o_lst);
    for(unsigned i_o = 0; i_o < o_lst.size(); i_o++)
    {
        AutoHD<TVArchivator> val = mod.at().valAt(o_lst[i_o]);
        if( val.at().startStat() )
            try{ val.at().stop(); }
            catch(TError err)
            {
                mess_err(err.cat.c_str(), "%s", err.mess.c_str());
                mess_err(nodePath().c_str(), _("Значення архіватора '%s' stop
error."), o_lst[i_o].c_str());
            }
    }
    // Повідомлення про зупинку архіватора
    mod.at().messList(o_lst);
    for(unsigned i_o = 0; i_o < o_lst.size(); i_o++)
    {
        AutoHD<TMArchivator> mess = mod.at().messAt(o_lst[i_o]);
        if( mess.at().startStat() )
            try{ mess.at().stop(); }
            catch(TError err)
            {
                mess_err(err.cat.c_str(), "%s", err.mess.c_str());
                mess_err(nodePath().c_str(), _("Повідомлення про помилку зупинки
архіватора"), o_lst[i_o].c_str());
            }
    }
}

//> Значення зупинки архіватора
vallList(o_lst);
for(unsigned i_o = 0; i_o < o_lst.size(); i_o++)
{
    AutoHD<TVArchive> aval = valAt(o_lst[i_o]);
    if( aval.at().startStat() )
        try{ aval.at().stop(); }
        catch(TError err)
        {
            mess_err(err.cat.c_str(), "%s", err.mess.c_str());
            mess_err(nodePath().c_str(), _("Значення помилки зупинки
архіватора"), o_lst[i_o].c_str());
        }
}
}

void TArchiveS::messPut( time_t tm, int utm, const string &categ, int8_t level,
const string &mess )
{
    //> Відправляємо повідомлення у буфер
    ResAlloc res(mRes, true);
    mBuf[headBuf].time = tm;
    mBuf[headBuf].utime = utm;
    mBuf[headBuf].categ = categ;
    mBuf[headBuf].level = (TMess::Type)abs(level);
    mBuf[headBuf].mess = mess;
    if( ++headBuf >= mBuf.size() ) headBuf = 0;
    //> Пеервіряємо, чи це не повідомлення архіватора
    if( headBuf == headLstread )
    {
        if( !(bufErr&0x01) )
        {

```

```

        bufErr |= 0x01;
        res.release();
        mess_err(nodePath().c_str(),_("Буфер заповнений. Останнє
повідомлення!"));
        res.request(true);
    }
    if( ++headLstread >= mBuf.size() ) headLstread = 0;
}
//> Перевіряємо швидкість заповнення буфера.
else if( headBuf-headLstread > messBufLen( )/2 )
{
    if( !(bufErr&0x02) )
    {
        bufErr |= 0x02;
        res.release();
        mess_warning(nodePath().c_str(),_("Повідомлення про т, що швидкість
заповнення буфера дуже висока!"));
        res.request(true);
    }
}
else bufErr = 0;

//> Обробка тривог. Для рівня менше 0 тривоги встановлено
map<string, TMess::SRec>::iterator p;
if( level < 0 ) mAlarms[categ] =
TMess::SRec(tm,utm,categ, (TMess::Type)abs(level),mess);
else if( (p=mAlarms.find(categ)) != mAlarms.end()) mAlarms.erase(p);
}

void TArchiveS::messPut( const vector<TMess::SRec> &recs )
{
    for(unsigned i_r = 0; i_r < recs.size(); i_r++)
        messPut(recs[i_r].time,recs[i_r].utime,recs[i_r].categ,recs[i_r].level,recs[i_r].mess);
}

void TArchiveS::messGet( time_t b_tm, time_t e_tm, vector<TMess::SRec> & recs,
const string &category, int8_t level, const string &arch, time_t upTo )
{
    recs.clear();

    ResAlloc res(mRes,false);
    if(!upTo) upTo = time(NULL)+STD_INTERF_TM;
    TRegExp re(category, "p");

    //> Отримуємо записи з буфера
    unsigned i_buf = headBuf;
    while(level >= 0 && (!arch.size() || arch==BUF_ARCH_NM) && time(NULL) <
upTo)
    {
        if(mBuf[i_buf].time >= b_tm && mBuf[i_buf].time != 0 && mBuf[i_buf].time
<= e_tm &&
            mBuf[i_buf].level >= level && re.test(mBuf[i_buf].categ))
            recs.push_back(mBuf[i_buf]);
        if(++i_buf >= mBuf.size()) i_buf = 0;
        if(i_buf == headBuf) break;
    }

    //> Отримуємо записи з архівів
    vector<string> t_lst, o_lst;
    modList(t_lst);
    for(unsigned i_t = 0; level >= 0 && i_t < t_lst.size(); i_t++)
    {
        at(t_lst[i_t]).at().messList(o_lst);
        for(unsigned i_o = 0; i_o < o_lst.size() && time(NULL) < upTo; i_o++)
        {
            AutoHD<TMArchivator> archtor = at(t_lst[i_t]).at().messAt(o_lst[i_o]);
            if(archtor.at().startStat() && (!arch.size() ||
arch==archtor.at().workId()))

```

```

        archtor.at().get(b_tm,e_tm,recs,category,level);
    }
}

//> Запит процесу тривоги
if(level < 0)
{
    vector< pair<int64_t,TMess::SRec* > > mb;
    for(map<string,TMess::SRec>::iterator p = mAlarms.begin(); p !=
mAlarms.end() && time(NULL) < upTo; p++)
        if((p->second.time >= b_tm || b_tm == e_tm) && p->second.time <= e_tm
&&
            p->second.level >= abs(level) && re.test(p->second.category)
            mb.push_back(pair<int64_t,TMess::SRec* >(FTM(p->second), &p-
>second)));
    sort(mb.begin(),mb.end());
    for(unsigned i_b = 0; i_b < mb.size(); i_b++)
recs.push_back(*mb[i_b].second);
}
}

time_t TArchiveS::messBeg( const string &arch )
{
    time_t rez = 0;
    ResAlloc res(mRes,false);
    if(arch.empty() || arch == BUF_ARCH_NM)
    {
        unsigned i_buf = headBuf;
        while(!arch.size() || arch == BUF_ARCH_NM)
        {
            rez = rez ? vmin(rez,mBuf[i_buf].time) : mBuf[i_buf].time;
            if(++i_buf >= mBuf.size()) i_buf = 0;
            if(i_buf == headBuf) break;
        }
        if( !arch.empty() ) return rez;
    }

    //- Отримуємо записи з архівів -
    vector<string> t_lst, o_lst;
    modList(t_lst);
    AutoHD<TMArchivator> archtor;
    for(unsigned i_t = 0; i_t < t_lst.size(); i_t++)
    {
        at(t_lst[i_t]).at().messList(o_lst);
        for(unsigned i_o = 0; i_o < o_lst.size(); i_o++)
        {
            archtor = at(t_lst[i_t]).at().messAt(o_lst[i_o]);
            if(archtor.at().startStat() && (!arch.size() ||
arch==archtor.at().workId()))
                rez = rez ? vmin(rez,archtor.at().begin()) : archtor.at().begin();
        }
    }

    return rez;
}

time_t TArchiveS::messEnd( const string &arch )
{
    time_t rez = 0;
    ResAlloc res(mRes,false);
    if(arch.empty() || arch == BUF_ARCH_NM)
    {
        unsigned i_buf = headBuf;
        while(!arch.size() || arch == BUF_ARCH_NM)
        {
            rez = rez ? vmax(rez,mBuf[i_buf].time) : mBuf[i_buf].time;
            if(++i_buf >= mBuf.size()) i_buf = 0;
            if(i_buf == headBuf) break;
        }
    }
}

```

```

    if(!arch.empty()) return rez;
}

//> Отримуюмо записи з архівів
vector<string> t_lst, o_lst;
modList(t_lst);
AutoHD<TMArchivator> archtor;
for(unsigned i_t = 0; i_t < t_lst.size(); i_t++)
{
    at(t_lst[i_t]).at().messList(o_lst);
    for(unsigned i_o = 0; i_o < o_lst.size(); i_o++)
    {
        archtor = at(t_lst[i_t]).at().messAt(o_lst[i_o]);
        if(archtor.at().startStat() && (!arch.size() ||
arch==archtor.at().workId()))
            rez = rez ? vmax(rez,archtor.at().end()) : archtor.at().end();
    }
}

return rez;
}

void TArchiveS::setMessBufLen(unsigned len)
{
    ResAlloc res(mRes,true);
    len = vmin(BUF_SIZE_MAX,vmax(BUF_SIZE_DEF,len));
    while(mBuf.size() > len)
    {
        mBuf.erase(mBuf.begin() + headBuf);
        if(headBuf >= mBuf.size()) headBuf = 0;
        if(headLstread >= mBuf.size()) headLstread = mBuf.size()-1;
    }
    while(mBuf.size() < len) mBuf.insert(mBuf.begin() + headBuf, TMess::SRec());
    modif();
}

void TArchiveS::setActValArch( const string &id, bool val )
{
    unsigned i_arch;

    ResAlloc res(vRes,true);
    for( i_arch = 0; i_arch < actUpSrc.size(); i_arch++ )
        if( actUpSrc[i_arch].at().id() == id ) break;

    if( val && i_arch >= actUpSrc.size() )
        actUpSrc.push_back(valAt(id));
    if( !val && i_arch < actUpSrc.size() )
        actUpSrc.erase(actUpSrc.begin()+i_arch);
}

void TArchiveS::setMessPeriod( int ivl )
{
    mMessPer = ivl;
    modif();

    if( subStartStat( ) )
    {
        struct itimerspec itval;
        itval.it_interval.tv_sec = itval.it_value.tv_sec = mMessPer;
        itval.it_interval.tv_nsec = itval.it_value.tv_nsec = 0;
        timer_settime(tmIdMess, 0, &itval, NULL);
    }
}

void TArchiveS::ArhMessTask( union signal obj )
{
    TArchiveS &arh = *(TArchiveS *)obj.sival_ptr;
    if( arh.prcStMess ) return;
    arh.prcStMess = true;
}

```

```

//> Читаємо повідомлення з буфера
if( arh.headLstread != arh.headBuf )
  try
  {
    ResAlloc res(arh.mRes,false);

    //> Беремо нове повідомлення
    unsigned new_headLstread = arh.headBuf;
    unsigned i_m = arh.headLstread;
    vector<TMess::SRec> o_mess;
    while( i_m != new_headLstread )
    {
      o_mess.push_back(arh.mBuf[i_m]);
      if( ++i_m >= arh.mBuf.size() ) i_m = 0;
    }
    arh.headLstread = new_headLstread;

    res.release();

    //> Архівуємо
    vector<string> t_lst, o_lst;
    arh.modList(t_lst);
    for(unsigned i_t = 0; i_t < t_lst.size(); i_t++)
    {
      arh.at(t_lst[i_t]).at().messList(o_lst);
      for(unsigned i_o = 0; i_o < o_lst.size(); i_o++)
        if(arh.at(t_lst[i_t]).at().messAt(o_lst[i_o]).at().startStat())
          arh.at(t_lst[i_t]).at().messAt(o_lst[i_o]).at().put(o_mess);
    }
  }
  catch(TError err)
  {
    mess_err(err.cat.c_str(),"%",err.mess.c_str());
    mess_err(arh.nodePath().c_str(),"(Помилка читання повідомлення з
буфера.)");
  }

  arh.prcStMess = false;
}

void *TArchiveS::ArhValTask( void *param )
{
  TArchiveS &arh = *(TArchiveS *)param;
  arh.endrunReqVal = false;
  arh.prcStVal = true;

  while( !arh.endrunReqVal )
  {
    int64_t work_tm = SYS->curTime();

    arh.vRes.resRequestR( );
    for(unsigned i_arh = 0; i_arh < arh.actUpSrc.size(); i_arh++)
      try
      {
        {
          if( work_tm/arh.actUpSrc[i_arh].at().period() >
arh.actUpSrc[i_arh].at().end()/arh.actUpSrc[i_arh].at().period() )
            arh.actUpSrc[i_arh].at().getActiveData();
        }
        catch(TError err)
        { mess_err(err.cat.c_str(),"%",err.mess.c_str()); }
        arh.vRes.resRelease( );

        TSYS::taskSleep((int64_t)arh.valPeriod()*1000000);
      }

    arh.prcStVal = false;

    return NULL;
  }
}

```

```

}

TVariant TArchiveS::objFuncCall( const string &iid, vector<TVariant> &prms,
const string &user )
{
    // Array messGet(int btm, int etm, string cat = "", int lev = 0, string arch
= ""); - запит системних повідомлень, для часу з <btm>
    // до <etm> для категорії <cat>, рівня <lev> та архіву <arch>
    // btm - час початку
    // etm - час закінчення
    // cat - категорія повідомлення
    // lev - рівні повідомлень
    // arch - архівація повідомлення
    if( iid == "messGet" && prms.size() >= 2 )
    {
        vector<TMess::SRec> recs;
        messGet( prms[0].getI(), prms[1].getI(), recs, ((prms.size())>=3) ?
prms[2].getS() : string(""),
            ((prms.size())>=4) ? prms[3].getI() : 0, ((prms.size())>=5) ?
prms[4].getS() : string("")) );
        TArrayObj *rez = new TArrayObj();
        for(unsigned i_m = 0; i_m < recs.size(); i_m++)
        {
            TVarObj *am = new TVarObj();
            am->propSet("tm", (int)recs[i_m].time);
            am->propSet("utm", recs[i_m].utime);
            am->propSet("categ", recs[i_m].categ);
            am->propSet("level", recs[i_m].level);
            am->propSet("mess", recs[i_m].mess);
            rez->propSet(TSYS::int2str(i_m), am);
        }
        return rez;
    }

    return TCntrNode::objFuncCall(iid, prms, user);
}

void TArchiveS::cntrCmdProc( XMLNode *opt )
{
    string a_path = opt->attr("path");
    //> Сервіс управління процесами
    if(a_path == "/serv/mess") //Повідомлення доступу
    {
        if(ctrChkNode(opt, "info", RWRWRW, "root", SARH_ID, SEC_RD))
        //Інформаційні повідомлення
        {
            string arch = opt->attr("arch");
            opt->setAttr("end", TSYS::uint2str(messEnd(arch)));
            opt->setAttr("beg", TSYS::uint2str(messBeg(arch)));
        }
        else if(ctrChkNode(opt, "get", RWRWRW, "root", SARH_ID, SEC_RD)) //Запит
значення дати
        {
            time_t tm = strtoul(opt->attr("tm").c_str(), 0, 10);
            time_t tm_grnd = strtoul(opt->attr("tm_grnd").c_str(), 0, 10);
            string arch = opt->attr("arch");
            string cat = opt->attr("cat");
            int lev = atoi(opt->attr("lev").c_str());
            vector<TMess::SRec> rez;
            messGet( tm_grnd, tm, rez, cat, (TMess::Type)lev, arch );
            for(unsigned i_r = 0; i_r < rez.size(); i_r++)
                opt->childAdd("el")->
                    setAttr("time", TSYS::uint2str(rez[i_r].time))->
                    setAttr("utime", TSYS::uint2str(rez[i_r].utime))->
                    setAttr("cat", rez[i_r].categ)->
                    setAttr("lev", TSYS::int2str(rez[i_r].level))->
                    setText(rez[i_r].mess);
        }
    }
    return;
}

```

```

}

//> Беремо сторінку інформації
if(opt->name() == "info")
{
    TSubSYS::cntrCmdProc(opt);
    ctrMkNode("grp",opt,-1,"/br/va_",_("Архів
значень"),RWRWR_,"root",SARH_ID,2,"idm","1","idSz","20");
    if(ctrMkNode("area",opt,1,"/m_arch",_("Архів
повідомлень"),R_R_R_,"root",SARH_ID))
    {
        ctrMkNode("fld",opt,-1,"/m_arch/size",_("Повідомлення розміру
буферу"),RWRWR_,"root",SARH_ID,2,"tp","dec","min",TSYS::int2str(BUF_SIZE_DEF).c_
str());
        ctrMkNode("fld",opt,-1,"/m_arch/per",_("Період архівування
(s)"),RWRWR_,"root",SARH_ID,1,"tp","dec");
        if(ctrMkNode("area",opt,-1,"/m_arch/view",_("Перегляд
повідомлень"),R_R___,"root",SARH_ID))
        {
            ctrMkNode("fld",opt,-
1,"/m_arch/view/tm",_("Time"),RWRW___,"root",SARH_ID,1,"tp","time");
            ctrMkNode("fld",opt,-1,"/m_arch/view/size",_("Розмір
(s)"),RWRW___,"root",SARH_ID,1,"tp","dec");
            ctrMkNode("fld",opt,-1,"/m_arch/view/cat",_("Зразок
категорії"),RWRW___,"root",SARH_ID,2,"tp","str","help",
_("Шаблон категорії повідомлень або регулярне вираження.\n"
_("Використовуйте тимчасові символи для групового
виділення:\n ' ' - будь-які підрядки;\n '?' - будь-які символи.\n"
_("Регулярне вираження складається з символів/'/
(/mod_(System|LogicLev)/)."));
            ctrMkNode("fld",opt,-
1,"/m_arch/view/lvl",_("Рівень"),RWRW___,"root",SARH_ID,4,"tp","dec","min","-
7","max","7",
            "help",_("Отримуємо повідомлення для рівня більшого і
дорівнюючого цьому."));
            ctrMkNode("fld",opt,-
1,"/m_arch/view/archtor",_("Архіватор"),RWRW___,"root",SARH_ID,4,"tp","str","dest
","select","select","/m_arch/lstAMess",
            "help",_("Архівація повідомлення.\n Не встановлюємо архіватор
для процесів у буфері та інших архіваторів.\nВстановлюємо '<buffer>' для процесів
у буфері ."));
            if(ctrMkNode("table",opt,-
1,"/m_arch/view/mess",_("Messages"),R_R___,"root",SARH_ID))
            {
                ctrMkNode("list",opt,-
1,"/m_arch/view/mess/0",_("Time"),R_R___,"root",SARH_ID,1,"tp","time");
                ctrMkNode("list",opt,-
1,"/m_arch/view/mess/0a",_("mcsec"),R_R___,"root",SARH_ID,1,"tp","dec");
                ctrMkNode("list",opt,-
1,"/m_arch/view/mess/1",_("Category"),R_R___,"root",SARH_ID,1,"tp","str");
                ctrMkNode("list",opt,-
1,"/m_arch/view/mess/2",_("Lev."),R_R___,"root",SARH_ID,1,"tp","dec");
                ctrMkNode("list",opt,-
1,"/m_arch/view/mess/3",_("Message"),R_R___,"root",SARH_ID,1,"tp","str");
            }
        }
    }
    if(ctrMkNode("area",opt,2,"/v_arch",_("Архів
значень"),R_R_R_,"root",SARH_ID))
    {
        ctrMkNode("fld",opt,-1,"/v_arch/per",_("Беремо дані періоду
(ms)"),RWRWR_,"root",SARH_ID,1,"tp","dec");
        ctrMkNode("fld",opt,-1,"/v_arch/prior",_("Беремо дані завдання
пріоритетного рівня"),RWRWR_,"root",SARH_ID,1,"tp","dec");
        ctrMkNode("fld",opt,-
1,"/v_arch/nmb",_("Number"),R_R_R_,"root",SARH_ID,1,"tp","str");
        ctrMkNode("list",opt,-1,"/v_arch/archs",_("Архів
значень"),RWRWR_,"root",SARH_ID,5,"tp","br","idm","1","s_com","add,del","br_pref
","va_","idSz","20");
    }
}

```

```

    }
    ctrMkNode("fld",opt,-1,"/help/g_help",_("Опції
допомоги"),R_R____,"root",SARH_ID,3,"tp","str","cols","90","rows","10");
    return;
}

//> Процес управління на сторінці
if(a_path == "/m_arch/per")
{
    if(ctrChkNode(opt,"get",RWRWR_,"root",SARH_ID,SEC_RD))    opt->
>setText(TSYS::int2str(messPeriod()));
    if(ctrChkNode(opt,"set",RWRWR_,"root",SARH_ID,SEC_WR))
    setMessPeriod(atoi(opt->text().c_str()));
}
else if(a_path == "/m_arch/size")
{
    if(ctrChkNode(opt,"get",RWRWR_,"root",SARH_ID,SEC_RD))    opt->
>setText(TSYS::int2str(messBufLen()));
    if(ctrChkNode(opt,"set",RWRWR_,"root",SARH_ID,SEC_WR))
    setMessBufLen(atoi(opt->text().c_str()));
}
else if(a_path == "/m_arch/view/tm")
{
    if(ctrChkNode(opt,"get",RWRW__, "root", SARH_ID, SEC_RD))
    {
        opt->setText(TBDS::genDBGet(nodePath()+"messTm","0",opt->
>attr("user")));
        if(!atoi(opt->text().c_str()))    opt->
>setText(TSYS::int2str(time(NULL)));
    }
    if(ctrChkNode(opt,"set",RWRW__, "root", SARH_ID, SEC_WR))
        TBDS::genDBSet(nodePath()+"messTm", (atoi(opt->
>text().c_str())>=time(NULL))?"0":opt->text(),opt->attr("user"));
}
else if(a_path == "/m_arch/view/size")
{
    if(ctrChkNode(opt,"get",RWRWR_,"root",SARH_ID,SEC_RD))    opt->
>setText(TBDS::genDBGet(nodePath()+"messSize","60",opt->attr("user")));
    if(ctrChkNode(opt,"set",RWRWR_,"root",SARH_ID,SEC_WR))
        TBDS::genDBSet(nodePath()+"messSize",opt->text(),opt->attr("user"));
}
else if(a_path == "/m_arch/view/cat")
{
    if(ctrChkNode(opt,"get",RWRW__, "root", SARH_ID, SEC_RD))    opt->
>setText(TBDS::genDBGet(nodePath()+"messCat","",opt->attr("user")));
    if(ctrChkNode(opt,"set",RWRW__, "root", SARH_ID, SEC_WR))
        TBDS::genDBSet(nodePath()+"messCat",opt->text(),opt->attr("user"));
}
else if(a_path == "/m_arch/view/lvl")
{
    if(ctrChkNode(opt,"get",RWRW__, "root", SARH_ID, SEC_RD))    opt->
>setText(TBDS::genDBGet(nodePath()+"messLev","0",opt->attr("user")));
    if(ctrChkNode(opt,"set",RWRW__, "root", SARH_ID, SEC_WR))
        TBDS::genDBSet(nodePath()+"messLev",opt->text(),opt->attr("user"));
}
else if(a_path == "/m_arch/view/archtor")
{
    if(ctrChkNode(opt,"get",RWRW__, "root", SARH_ID, SEC_RD))    opt->
>setText(TBDS::genDBGet(nodePath()+"messArch","",opt->attr("user")));
    if(ctrChkNode(opt,"set",RWRW__, "root", SARH_ID, SEC_WR))
        TBDS::genDBSet(nodePath()+"messArch",opt->text(),opt->attr("user"));
}
else if(a_path == "/m_arch/lstAMess" && ctrChkNode(opt,"get",R_R____))
{
    opt->childAdd("el")->setText("");
    opt->childAdd("el")->setText(BUF_ARCH_NM);
    vector<string> lsm, lsa;
    modList(lsm);
    for(unsigned i_m = 0; i_m < lsm.size(); i_m++)

```

```

    {
        at(lsm[i_m]).at().messList(lsa);
        for(unsigned i_a = 0; i_a < lsa.size(); i_a++)
            opt->childAdd("el")->setText(lsm[i_m]+"."+lsa[i_a]);
    }
}
else if(a_path == "/m_arch/view/mess" &&
ctrChkNode(opt,"get",R_R____,"root",SARH_ID))
{
    vector<TMess::SRec> rec;
    time_t gtm = atoi(TBDS::genDBGet(nodePath()+"messTm","0",opt-
>attr("user")).c_str());
    if(!gtm) gtm = time(NULL);
    int gsz = atoi(TBDS::genDBGet(nodePath()+"messSize","60",opt-
>attr("user")).c_str());
    messGet(gtm-gsz, gtm, rec,
            TBDS::genDBGet(nodePath()+"messCat","",opt->attr("user")),
            atoi(TBDS::genDBGet(nodePath()+"messLev","0",opt-
>attr("user")).c_str()),
            TBDS::genDBGet(nodePath()+"messArch","",opt->attr("user")) );

    XMLNode *n_tm = ctrMkNode("list",opt,-
1,"/m_arch/view/mess/0","",R_R____,"root",SARH_ID);
    XMLNode *n_tmu = ctrMkNode("list",opt,-
1,"/m_arch/view/mess/0a","",R_R____,"root",SARH_ID);
    XMLNode *n_cat = ctrMkNode("list",opt,-
1,"/m_arch/view/mess/1","",R_R____,"root",SARH_ID);
    XMLNode *n_lvl = ctrMkNode("list",opt,-
1,"/m_arch/view/mess/2","",R_R____,"root",SARH_ID);
    XMLNode *n_mess = ctrMkNode("list",opt,-
1,"/m_arch/view/mess/3","",R_R____,"root",SARH_ID);
    for(int i_rec = rec.size()-1; i_rec >= 0; i_rec--)
    {
        if(n_tm) n_tm->childAdd("el")-
>setText(TSYS::int2str(rec[i_rec].time));
        if(n_tmu) n_tmu->childAdd("el")-
>setText(TSYS::int2str(rec[i_rec].utime));
        if(n_cat) n_cat->childAdd("el")->setText(rec[i_rec].categ);
        if(n_lvl) n_lvl->childAdd("el")-
>setText(TSYS::int2str(rec[i_rec].level));
        if(n_mess) n_mess->childAdd("el")->setText(rec[i_rec].mess);
    }
}
else if(a_path == "/v_arch/per")
{
    if(ctrChkNode(opt,"get",RWRWR_,"root",SARH_ID,SEC_RD)) opt-
>setText(TSYS::int2str(valPeriod()));
    if(ctrChkNode(opt,"set",RWRWR_,"root",SARH_ID,SEC_WR))
        setValPeriod(atoi(opt->text().c_str()));
}
else if(a_path == "/v_arch/prior")
{
    if(ctrChkNode(opt,"get",RWRWR_,"root",SARH_ID,SEC_RD)) opt-
>setText(TSYS::int2str(valPrior()));
    if(ctrChkNode(opt,"set",RWRWR_,"root",SARH_ID,SEC_WR))
        setValPrior(atoi(opt->text().c_str()));
}
else if(a_path == "/v_arch/nmb" && ctrChkNode(opt))
{
    vector<string> list;
    vallist(list);
    unsigned e_c = 0;
    for(unsigned i_a = 0; i_a < list.size(); i_a++)
        if(valAt(list[i_a]).at().startStat()) e_c++;
    opt->setText(TSYS::strMess_("All: %d; Enabled: %d"),list.size(),e_c);
}
else if(a_path == "/br/va_" || a_path == "/v_arch/archs")
{
    if(ctrChkNode(opt,"get",RWRWR_,"root",SARH_ID,SEC_RD))

```

```

    {
        vector<string> list;
        valList(list);
        for(unsigned i_a=0; i_a < list.size(); i_a++)
            opt->childAdd("el")->setAttr("id",list[i_a])-
>setText(valAt(list[i_a]).at().name());
    }
    if(ctrChkNode(opt,"add",RWRWR_,"root",SARH_ID,SEC_WR)
    {
        string vid = TSYS::strEncode(opt->attr("id"),TSYS::oscdID);
        valAdd(vid); valAt(vid).at().setName(opt->text());
    }
    if(ctrChkNode(opt,"del",RWRWR_,"root",SARH_ID,SEC_WR)
    chldDel(mAval,opt->attr("id"),-1,1);
    }
    else if(a_path == "/help/g_help" &&
ctrChkNode(opt,"get",R_R___,"root",SARH_ID)    opt->setText(optDescr());
    else TSubSYS::cntrCmdProc(opt);
    }

//*****
/* TTipArchivator *
//*****
TTipArchivator::TTipArchivator( const string &id ) : TModule(id)
{
    mVal = grpAdd("val_");
    mMess = grpAdd("mess_");
}

TTipArchivator::~~TTipArchivator()
{
    nodeDelAll();
}

TArchiveS &TTipArchivator::owner()
{
    return (TArchiveS &)TModule::owner();
}

void TTipArchivator::messAdd(const string &name, const string &idb )
{
    chldAdd(mMess, AMess(name,idb));
}

void TTipArchivator::valAdd( const string &iid, const string &idb )
{
    chldAdd(mVal, AVal(iid,idb));
}

void TTipArchivator::cntrCmdProc( XMLNode *opt )
{
    //> Беремо сторінку інформації
    if(opt->name() == "info")
    {
        TModule::cntrCmdProc(opt);
        ctrMkNode("area",opt,0,"/arch",_("Архіватори"));
        ctrMkNode("grp",opt,-1,"/br/mess_",_("Повідомлення
архіватора"),RWRWR_,"root",SARH_ID,2,"idm","1","idSz","20");
        ctrMkNode("grp",opt,-1,"/br/val_",_("Значення
архіватора"),RWRWR_,"root",SARH_ID,2,"idm","1","idSz","20");
        ctrMkNode("list",opt,-1,"/arch/mess",_("Повідомлення
архіваторів"),RWRWR_,"root",SARH_ID,5,"tp","br","idm","1","s_com","add,del","br_
pref","mess_","idSz","20");
        ctrMkNode("list",opt,-1,"/arch/val",_("Значення
архіваторів"),RWRWR_,"root",SARH_ID,5,"tp","br","idm","1","s_com","add,del","br_
pref","val_","idSz","20");
        return;
    }
    //> Процес управління на сторінці

```

```

string a_path = opt->attr("path");
if(a_path == "/br/mess_" || a_path == "/arch/mess")
{
    if(ctrChkNode(opt,"get",RWRWR_,"root",SARH_ID,SEC_RD)
    {
        vector<string> list;
        messList(list);
        for( unsigned i_a=0; i_a < list.size(); i_a++ )
            opt->childAdd("el")->setAttr("id",list[i_a])-
>setText(messAt(list[i_a]).at().name());
    }
    if(ctrChkNode(opt,"add",RWRWR_,"root",SARH_ID,SEC_WR)
    {
        string vid = TSYS::strEncode(opt->attr("id"),TSYS::oscdID);
        messAdd(vid); messAt(vid).at().setName(opt->text());
    }
    if(ctrChkNode(opt,"del",RWRWR_,"root",SARH_ID,SEC_WR)      messDel(opt-
>attr("id"),true);
    }
    else if(a_path == "/br/val_" || a_path == "/arch/val")
    {
        if(ctrChkNode(opt,"get",RWRWR_,"root",SARH_ID,SEC_RD)
        {
            vector<string> list;
            valList(list);
            for(unsigned i_a=0; i_a < list.size(); i_a++)
                opt->childAdd("el")->setAttr("id",list[i_a])-
>setText(valAt(list[i_a]).at().name());
        }
        if(ctrChkNode(opt,"add",RWRWR_,"root",SARH_ID,SEC_WR)
        {
            string vid = TSYS::strEncode(opt->attr("id"),TSYS::oscdID);
            valAdd(vid); valAt(vid).at().setName(opt->text());
        }
        if(ctrChkNode(opt,"del",RWRWR_,"root",SARH_ID,SEC_WR)      valDel(opt-
>attr("id"),true);
        }
        else TModule::cntrCmdProc(opt);
    }
}

//*****
//* Повідомлення архіватора *
//*****

//*****
//* TMArchivator *
//*****
TMArchivator::TMArchivator(const string &iid, const string &idb, TElem *cf_el) :
    TConfig( cf_el ), run_st(false),
    m_id(cfg("ID").getSd()), m_name(cfg("NAME").getSd()),
m_dscr(cfg("DESCR").getSd()), m_addr(cfg("ADDR").getSd()),
    m_cat_o(cfg("CATEG").getSd()), m_start(cfg("START").getBd()),
m_level(cfg("LEVEL").getId()), m_db(idb)
{
    m_id = iid;
}

TCntrNode &TMArchivator::operator=( TCntrNode &node )
{
    TMArchivator *src_n = dynamic_cast<TMArchivator*>(&node);
    if( !src_n ) return *this;

    //> Конфігурація копіювання
    string tid = id();
    *(TConfig*)this = *(TConfig*)src_n;
    cfg("MODUL").setS(owner().modId());
    m_id = tid;
    m_db = src_n->m_db;
}

```

```

    if( src_n->startStat() && toStart() && !startStat() )
        start( );

    return *this;
}

void TMArchivator::postEnable( int flag )
{
    cfg("MODUL").setS(owner().modId());
}

void TMArchivator::preDisable( int flag )
{
    if( startStat() ) stop( );
}

void TMArchivator::postDisable(int flag)
{
    try
    {
        if( flag )
            SYS->db().at().dataDel(fullDB(),SYS-
>archive().at().nodePath()+tbl(),*this,true);
    }catch(TError err)
    { mess_warning(err.cat.c_str(),"%s",err.mess.c_str()); }
}

TTipArchivator &TMArchivator::owner( )    { return *(TTipArchivator*)nodePrev(); }

string TMArchivator::workId( )            { return owner().modId()+"."+id(); }

string TMArchivator::name( )              { return (m_name.size())?m_name:m_id; }

string TMArchivator::tbl( )                { return
owner().owner().subId()+"_mess_proc"; }

void TMArchivator::load_( )
{
    if( !SYS->chkSelDB(DB()) ) return;
    SYS->db().at().dataGet(fullDB(),SYS->archive().at().nodePath()+tbl(),*this);
}

void TMArchivator::save_( )
{
    SYS->db().at().dataSet(fullDB(),SYS->archive().at().nodePath()+tbl(),*this);
}

void TMArchivator::categ( vector<string> &list )
{
    list.clear();
    string c_vl;
    for( int i_off = 0; (c_vl=TSYS::strSepParse(m_cat_o,0,',',&i_off)).size(); )
        list.push_back(c_vl);
}

bool TMArchivator::chkMessOK( const string &icateg, TMess::Type ilvl )
{
    vector<string> cat_ls;

    categ(cat_ls);

    if(ilvl >= level())
        for(unsigned i_cat = 0; i_cat < cat_ls.size(); i_cat++)
            if(TRegExp(cat_ls[i_cat], "p").test(icateg))
                return true;
    return false;
}

```

```

TVariant TMArchivator::objFuncCall( const string &iid, vector<TVariant> &prms,
const string &user )
{
    // bool status() - беремо статус архівування.
    if(iid == "status") return startStat();
    // int end() - беремо дані архівування, час закінчення.
    if(iid == "end") return (int)end();
    // int begin() - беремо дані архівування, час початку.
    if(iid == "begin") return (int)begin();

    //> Виклик функцій конфігурації
    TVariant cfRez = objFunc(iid, prms, user);
    if(!cfRez.isNull()) return cfRez;

    return TCntrNode::objFuncCall(iid, prms, user);
}

void TMArchivator::cntrCmdProc( XMLNode *opt )
{
    //> Беремо сторінку інформації
    if(opt->name() == "info")
    {
        TCntrNode::cntrCmdProc(opt);
        ctrMkNode("Cisco_Application_Centric_Infrastructure_cntr",opt,-
1, "/", _("Повідомлення архіватора: ") + name(), RWRWR_, "root", SARH_ID);
        if(ctrMkNode("area",opt,-1, "/prm", _("Архіватор")))
        {
            if(ctrMkNode("area",opt,-1, "/prm/st", _("Стан")))
            {
                ctrMkNode("fld",opt,-
1, "/prm/st/st", _("Running"), RWRWR_, "root", SARH_ID, 1, "tp", "bool");
                ctrMkNode("fld",opt,-1, "/prm/st/db", _("Архіватор
БД"), RWRWR_, "root", "root", 4,
"tp", "str", "dest", "select", "select", "/db/list", "help", TMess::labDB());
                ctrMkNode("fld",opt,-
1, "/prm/st/end", _("End"), R_R_R_, "root", "root", 1, "tp", "time");
                ctrMkNode("fld",opt,-
1, "/prm/st/beg", _("Begin"), R_R_R_, "root", "root", 1, "tp", "time");
            }
            if(ctrMkNode("area",opt,-1, "/prm/cfg", _("Конфігурація")))
            {
                ctrMkNode("fld",opt,-
1, "/prm/cfg/id", cfg("ID").fld().descr(), R_R_R_, "root", SARH_ID, 1, "tp", "str");
                ctrMkNode("fld",opt,-
1, "/prm/cfg/nm", cfg("NAME").fld().descr(), RWRWR_, "root", SARH_ID, 2, "tp", "str", "le
n", "50");
                ctrMkNode("fld",opt,-
1, "/prm/cfg/dscr", cfg("DESCR").fld().descr(), RWRWR_, "root", SARH_ID, 3, "tp", "str",
"cols", "90", "rows", "3");
                ctrMkNode("fld",opt,-
1, "/prm/cfg/addr", cfg("ADDR").fld().descr(), RWRWR_, "root", SARH_ID, 1, "tp", "str");
                ctrMkNode("fld",opt,-
1, "/prm/cfg/lvl", cfg("LEVEL").fld().descr(), RWRWR_, "root", SARH_ID, 2, "tp", "dec",
"help", _("Отримуємо повідомлення для рівня більшого і
дорівнюючого цьому."));
                ctrMkNode("fld",opt,-
1, "/prm/cfg/cats", cfg("CATEG").fld().descr(), RWRWR_, "root", SARH_ID, 2, "tp", "str",
"help", _("Шаблон категорії повідомлень або регулярне вираження у
процесі архівування, відокремлюються символом';'.\n"
"Використовуйте тимчасові символи для групового
виділення:\n '*' - будь-які підрядки;\n '?' - будь-які символи.\n"
"Регулярне вираження складається з символів'/')
(/mod_(System|LogicLev)/)."));
                ctrMkNode("fld",opt,-1, "/prm/cfg/start", _("To
start"), RWRWR_, "root", SARH_ID, 1, "tp", "bool");
            }
        }
    }
}

```

```

    if(run_st && ctrMkNode("area",opt,-
1, "/mess",_("Messages"),R_R___,"root",SARH_ID)
    {
        ctrMkNode("fld",opt,-
1, "/mess/tm",_("Time"),RWRW___,"root",SARH_ID,1,"tp","time");
        ctrMkNode("fld",opt,-1, "/mess/size",_("Розмір
(s)"),RWRW___,"root",SARH_ID,1,"tp","dec");
        ctrMkNode("fld",opt,-1, "/mess/cat",_("Зразок
катерорії"),RWRW___,"root",SARH_ID,2,"tp","str","help",
        _("Шаблон катерорії повідомлень або регулярне вираження.\n"
        "Використовуйте тимчасові символи для групового виділення:\n
'*' - будь-які підрядки;\n '?' - будь-які символи.\n"
        "Регулярне вираження складається з символів'/
(/mod_(System|LogicLev)/)."));
        ctrMkNode("fld",opt,-
1, "/mess/lvl",_("Level"),RWRW___,"root",SARH_ID,4,"tp","dec","min","0","max","7",
        "help",_("Отримуємо повідомлення для рівня більшого і дорівнюючого
цьому."));
        if(ctrMkNode("table",opt,-
1, "/mess/mess",_("Messages"),R_R___,"root",SARH_ID)
        {
            ctrMkNode("list",opt,-
1, "/mess/mess/0",_("Час"),R_R___,"root",SARH_ID,1,"tp","time");
            ctrMkNode("list",opt,-
1, "/mess/mess/0a",_("mcsec"),R_R___,"root",SARH_ID,1,"tp","dec");
            ctrMkNode("list",opt,-
1, "/mess/mess/1",_("Катерорія"),R_R___,"root",SARH_ID,1,"tp","str");
            ctrMkNode("list",opt,-
1, "/mess/mess/2",_("Рівень"),R_R___,"root",SARH_ID,1,"tp","dec");
            ctrMkNode("list",opt,-
1, "/mess/mess/3",_("Повідомлення"),R_R___,"root",SARH_ID,1,"tp","str");
        }
        return;
    }
    //> Процес управління на сторінці
    string a_path = opt->attr("path");
    if(a_path == "/prm/st/st")
    {
        if(ctrChkNode(opt,"get",RWRWR_,"root",SARH_ID,SEC_RD)
startStat() ? "1" : "0" );
        if(ctrChkNode(opt,"set",RWRWR_,"root",SARH_ID,SEC_WR)
atoi(opt-
>text().c_str()) ? start() : stop());
    }
    else if(a_path == "/prm/st/db")
    {
        if(ctrChkNode(opt,"get",RWRWR_,"root",SARH_ID,SEC_RD)
opt->setText(
DB() );
        if(ctrChkNode(opt,"set",RWRWR_,"root",SARH_ID,SEC_WR)
setDB( opt-
>text() );
    }
    else if(a_path == "/prm/st/end" && ctrChkNode(opt))
opt->setText(
TSYS::int2str(end()) );
    else if(a_path == "/prm/st/beg" && ctrChkNode(opt))
opt->setText(
TSYS::int2str(begin()) );
    else if(a_path == "/prm/cfg/id" && ctrChkNode(opt))
opt->setText(
id() );
    else if(a_path == "/prm/cfg/nm" )
    {
        if(ctrChkNode(opt,"get",RWRWR_,"root",SARH_ID,SEC_RD)
opt->setText(
name() );
        if(ctrChkNode(opt,"set",RWRWR_,"root",SARH_ID,SEC_WR)
setName( opt-
>text() );
    }
    else if(a_path == "/prm/cfg/dscr")
    {
        if(ctrChkNode(opt,"get",RWRWR_,"root",SARH_ID,SEC_RD)
opt->setText(
dscr() );
    }

```

```

        if(ctrChkNode(opt,"set",RWRWR_,"root",SARH_ID,SEC_WR))      setDscr( opt-
>text() );
    }
    else if(a_path == "/prm/cfg/addr")
    {
        if(ctrChkNode(opt,"get",RWRWR_,"root",SARH_ID,SEC_RD))      opt->setText(
addr() );
        if(ctrChkNode(opt,"set",RWRWR_,"root",SARH_ID,SEC_WR))      setAddr( opt-
>text() );
    }
    else if(a_path == "/prm/cfg/lvl")
    {
        if(ctrChkNode(opt,"get",RWRWR_,"root",SARH_ID,SEC_RD))      opt-
>setText(TSYS::int2str(level()));
        if(ctrChkNode(opt,"set",RWRWR_,"root",SARH_ID,SEC_WR))      setLevel(
atoi(opt->text().c_str()));
    }
    else if(a_path == "/prm/cfg/start")
    {
        if(ctrChkNode(opt,"get",RWRWR_,"root",SARH_ID,SEC_RD))      opt->setText(
toStart() ? "1" : "0" );
        if(ctrChkNode(opt,"set",RWRWR_,"root",SARH_ID,SEC_WR))      setToStart(
atoi(opt->text().c_str()));
    }
    else if(a_path == "/prm/cfg/cats")
    {
        if(ctrChkNode(opt,"get",RWRWR_,"root",SARH_ID,SEC_RD))      opt-
>setText(m_cat_o);
        if(ctrChkNode(opt,"set",RWRWR_,"root",SARH_ID,SEC_WR))      { m_cat_o =
opt->text(); modif(); }
    }
    else if(a_path == "/mess/tm")
    {
        if(ctrChkNode(opt,"get",RWRW__, "root", SARH_ID, SEC_RD))
        {
            opt->setText(TBDS::genDBGet(nodePath()+"messTm","0",opt-
>attr("user")));
            if(!atoi(opt->text().c_str())) opt-
>setText(TSYS::int2str(time(NULL)));
        }
        if(ctrChkNode(opt,"set",RWRW__, "root", SARH_ID, SEC_WR))
            TBDS::genDBSet(nodePath()+"messTm", (atoi(opt-
>text().c_str())>time(NULL))?"0":opt->text(),opt->attr("user"));
    }
    else if(a_path == "/mess/size")
    {
        if(ctrChkNode(opt,"get",RWRW__, "root", SARH_ID, SEC_RD))      opt-
>setText(TBDS::genDBGet(nodePath()+"messSize","10",opt->attr("user")));
        if(ctrChkNode(opt,"set",RWRW__, "root", SARH_ID, SEC_WR))
            TBDS::genDBSet(nodePath()+"messSize",opt->text(),opt->attr("user"));
    }
    else if(a_path == "/mess/cat")
    {
        if(ctrChkNode(opt,"get",RWRW__, "root", SARH_ID, SEC_RD))      opt-
>setText(TBDS::genDBGet(nodePath()+"messCat","",opt->attr("user")));
        if(ctrChkNode(opt,"set",RWRW__, "root", SARH_ID, SEC_WR))
            TBDS::genDBSet(nodePath()+"messCat",opt->text(),opt->attr("user"));
    }
    else if(a_path == "/mess/lvl")
    {
        if(ctrChkNode(opt,"get",RWRW__, "root", SARH_ID, SEC_RD))      opt-
>setText(TBDS::genDBGet(nodePath()+"messLev","0",opt->attr("user")));
        if(ctrChkNode(opt,"set",RWRW__, "root", SARH_ID, SEC_WR))
            TBDS::genDBSet(nodePath()+"messLev",opt->text(),opt->attr("user"));
    }
    else if(a_path == "/mess/mess" && run_st &&
ctrChkNode(opt,"get",R_R____,"root",SARH_ID))
    {
        vector<TMess::SRec> rec;

```

```

        time_t end = atoi(TBDS::genDBGet(nodePath()+"messTm","0",opt-
>attr("user")).c_str());
        if( !end ) end = time(NULL);
        time_t beg = end - atoi(TBDS::genDBGet(nodePath()+"messSize","10",opt-
>attr("user")).c_str());
        string cat = TBDS::genDBGet(nodePath()+"messCat","",opt->attr("user"));
        char lev = atoi(TBDS::genDBGet(nodePath()+"messLev","0",opt-
>attr("user")).c_str());

        get( beg, end, rec, cat, lev );

        XMLNode *n_tm      = ctrMkNode("list",opt,-
1, "/mess/mess/0","",R_R____,"root",SARH_ID);
        XMLNode *n_tmu     = ctrMkNode("list",opt,-
1, "/mess/mess/0a","",R_R____,"root",SARH_ID);
        XMLNode *n_cat     = ctrMkNode("list",opt,-
1, "/mess/mess/1","",R_R____,"root",SARH_ID);
        XMLNode *n_lvl     = ctrMkNode("list",opt,-
1, "/mess/mess/2","",R_R____,"root",SARH_ID);
        XMLNode *n_mess    = ctrMkNode("list",opt,-
1, "/mess/mess/3","",R_R____,"root",SARH_ID);
        for(int i_rec = rec.size()-1; i_rec >= 0; i_rec--)
        {
            if(n_tm)        n_tm->childAdd("el")-
>setText(TSYS::int2str(rec[i_rec].time));
            if(n_tmu)       n_tmu->childAdd("el")-
>setText(TSYS::int2str(rec[i_rec].utime));
            if(n_cat)       n_cat->childAdd("el")->setText(rec[i_rec].categ);
            if(n_lvl)       n_lvl->childAdd("el")-
>setText(TSYS::int2str(rec[i_rec].level));
            if(n_mess)      n_mess->childAdd("el")->setText(rec[i_rec].mess);
        }
    }
    else TCntrNode::cntrCmdProc(opt);
}

```

tmodule.cpp - робота з модулями

```

//WebCisco_Application_Centric_Infrastructure системний файл: tmodule.cpp

#include <sys/types.h>
#include <sys/stat.h>
#include <stdarg.h>
#include <unistd.h>
#include <dlfcn.h>
#include <string.h>
#include <libintl.h>

#include "tsys.h"
#include "terror.h"
#include "tmess.h"
#include "tsubsys.h"
#include "tmodule.h"

using namespace Cisco_Application_Centric_Infrastructure;

//*****
/* TModule
//*****
const char *TModule::l_info[] =
    {"Модуль", "Ім'я", "Тип", "Джерело", "Версія", "Автор", "Дескриптор", "Ліцензія"};

TModule::TModule( const string &id ) : mId(id)
{
    lc_id = string("oscd_")+mId;
    bindtextdomain(lc_id.c_str(), LOCALEDIR);

    //> Переведення динамічного рядка
#ifdef 0
    char mess[][100] = { _("Автор"), _("Ліцензія") };
#endif
}

TModule::~TModule( )
{
    //> Очищення списку експортуємих функцій
    for(unsigned i = 0; i < m_efunc.size(); i++)
        delete m_efunc[i];
}

string TModule::modName()
{
    return mName;
}

void TModule::postEnable( int flag )
{
    if(flag&TCntrNode::NodeRestore) return;

    mess_info(nodePath().c_str(), _("З'єднання з модулем!"));
}

TSubSYS &TModule::owner( )    { return *(TSubSYS*)nodePrev(); }

void TModule::modFuncList( vector<string> &list )
{
    list.clear();
    for(unsigned i = 0; i < m_efunc.size(); i++)
        list.push_back(m_efunc[i]->prot);
}

bool TModule::modFuncPresent( const string &prot )
{
    for(unsigned i = 0; i < m_efunc.size(); i++)

```

```

        if(m_efunc[i]->prot == prot)
            return true;
        return false;
    }

TModule::ExpFunc &TModule::modFunc( const string &prot )
{
    for(unsigned i = 0; i < m_efunc.size(); i++)
        if(m_efunc[i]->prot == prot) return *m_efunc[i];
    throw TError(nodePath().c_str(),_("Функція '%s' не представлена у
модулі!"),prot.c_str());
}

void TModule::modFunc( const string &prot, void (TModule::*offptr)() )
{
    *offptr = modFunc(prot).ptr;
}

void TModule::modInfo( vector<string> &list )
{
    for(unsigned i_opt = 0; i_opt < sizeof(l_info)/sizeof(char *); i_opt++)
        list.push_back(l_info[i_opt]);
}

string TModule::modInfo( const string &name )
{
    string info;

    if(name == l_info[0]) info = mId;
    else if(name == l_info[1]) info = mName;
    else if(name == l_info[2]) info = mType;
    else if(name == l_info[3]) info = mSource;
    else if(name == l_info[4]) info = mVers;
    else if(name == l_info[5]) info = mAuthor;
    else if(name == l_info[6]) info = mDescr;
    else if(name == l_info[7]) info = mLicense;

    return info;
}

void TModule::cntrCmdProc( XMLNode *opt )
{
    //> Беремо сторінку інформації
    if(opt->name() == "info")
    {
        TCntrNode::cntrCmdProc(opt);
        ctrMkNode("Cisco_Application_Centric_Infrastructure_cntr",opt,-
1, "/", _("Module: ") + modId(), R_R_R_);
        ctrMkNode("branches",opt,-1, "/br", "", R_R_R_);
        if(TUIS::icoPresent(owner().subId() + "." + modId())) ctrMkNode("img",opt,-
1, "/ico", "", R_R_R_);
        if(ctrMkNode("area",opt,-1, "/help", _("Help")))
            if(ctrMkNode("area",opt,-1, "/help/m_inf", _("Модуль інформації")))
            {
                vector<string> list;
                modInfo(list);
                for(unsigned i_l = 0; i_l < list.size(); i_l++)
                    ctrMkNode("fld",opt,-
1, (string("/help/m_inf/") + list[i_l]).c_str(),_(list[i_l].c_str()),R_R_R_,"root",
"root",1,"tp","str");
            }
        return;
    }

    //> Процес управління на сторінці
    string a_path = opt->attr("path");
    if(a_path == "/ico" && ctrChkNode(opt))
    {
        string itp;

```

```
    opt-
>setText(TSYS::strEncode(TUIS::icoGet(owner().subId()+".")+modId(),&itp),TSYS::ba
se64));
    opt->setAttr("tp",itp);
    }
    else if(a_path.substr(0,11) == "/help/m_inf" && ctrChkNode(opt))
        opt->setText(modInfo(TSYS::pathLev(a_path,2)));
    else TCntrNode::cntrCmdProc(opt);
}

const char *TModule::I18N( const char *mess )
{
    const char *rez = Mess->I18N(mess,lc_id.c_str());
    if( !strcmp(mess,rez) ) rez = _(mess);
    return rez;
}
```

К6П3_2024

tparamcontr.cpp - параметри управління системою

```

//WebCisco_Application_Centric_Infrastructure системний файл: tparamcontr.cpp

#include "tbds.h"
#include "tsys.h"
#include "tmess.h"
#include "tdaqs.h"
#include "tcontroller.h"
#include "ttipdaq.h"
#include "ttipparam.h"
#include "tparamcontr.h"

using namespace Cisco_Application_Centric_Infrastructure;

//*****
//* TParamContr *
//*****
TParamContr::TParamContr( const string &name, TTipParam *tpprm ) :
TConfig(tpprm, m_en(false), tipparm(tpprm)
{
    setId(name);
    setName(name);
}

TParamContr::~TParamContr( )
{
    nodeDelAll();
}

TCntrNode &TParamContr::operator=( TCntrNode &node )
{
    TParamContr *src_n = dynamic_cast<TParamContr*>(&node);
    if(!src_n) return *this;

    //> Перевіряємо тип параметрів й змінюємо їх, якщо вони змінні, або нижчі
    if(type().name != src_n->type().name && owner().owner().tpPrmToId(src_n-
>type().name) >= 0)
    {
        if(enableStat()) disable();
        setType(src_n->type().name);
    }

    //> Конфігурація копіювання
    string tid = id();
    *(TConfig*)this = *(TConfig*)src_n;
    setId(tid);

    //> Дозволяємо нові параметри
    if(src_n->enableStat() && toEnable( ) && !enableStat())enable();

    return *this;
}

TController &TParamContr::owner( ) { return *(TController*)nodePrev(); }

string TParamContr::name( ) { string nm = cfg("NAME").getS(); return nm.size()
? nm : id(); }

void TParamContr::setName( const string &inm ) { cfg("NAME").setS(inm); }

string TParamContr::descr( ) { return cfg("DESCR").getS(); }

void TParamContr::setDescr( const string &idsc ){ cfg("DESCR").setS(idsc); }

void TParamContr::postEnable(int flag)
{
    TValue::postEnable(flag);
}

```

```

    if(!vlCfg())    setVlCfg(this);
    if(!vlElemPresent(&SYS->daq().at().errE()))
        vlElemAtt(&SYS->daq().at().errE());
}

void TParamContr::preDisable(int flag)
{
    //> Видаляємо або зупиняємо архівування
    vector<string> a_ls;
    vlList(a_ls);

    for(unsigned i_a = 0; i_a < a_ls.size(); i_a++)
        if(!vlAt(a_ls[i_a]).at().arch().freeStat())
            {
                string arh_id = vlAt(a_ls[i_a]).at().arch().at().id();
                if(flag) SYS->archive().at().valDel(arh_id,true);
                else SYS->archive().at().valAt(arh_id).at().stop();
            }

    if(enableStat())    disable();
}

void TParamContr::postDisable(int flag)
{
    if(flag)
    {
        //> Видаляємо параметри з БД
        try
        {
            SYS->db().at().dataDel(owner().DB()+"."+owner().cfg(type().db).getS(),
owner().owner().nodePath()+owner().cfg(type().db).getS(),*this,true);
            }catch(TError err) { mess_err(err.cat.c_str(),"%s",err.mess.c_str()); }
        }
    }

void TParamContr::load_( )
{
    if(!SYS->chkSelDB(owner().DB())) return;

    cfgViewAll(true);
    SYS->db().at().dataGet(owner().DB()+"."+owner().cfg(type().db).getS(),
owner().owner().nodePath()+owner().cfg(type().db).getS(),*this);
}

void TParamContr::save_( )
{
    SYS->db().at().dataSet( owner().DB()+"."+owner().cfg(type().db).getS(),
owner().owner().nodePath()+owner().cfg(type().db).getS(),*this );

    //> Зберігаємо архіви
    vector<string> a_ls;
    vlList(a_ls);
    for(unsigned i_a = 0; i_a < a_ls.size(); i_a++)
        if(!vlAt(a_ls[i_a]).at().arch().freeStat())
            vlAt(a_ls[i_a]).at().arch().at().save();
}

bool TParamContr::cfgChange( TCfg &cfg ) { modif( ); return true; }

TParamContr & TParamContr::operator=( TParamContr & PrmCntr )
{
    TConfig::operator=(PrmCntr);

    return *this;
}

```

```

void TParamContr::enable()
{
    m_en = true;
}

void TParamContr::disable()
{
    m_en = false;
}

void TParamContr::vlGet( TVal &val )
{
    if( val.name() == "err" )
    {
        if( !enableStat() ) val.setS(_("1:Параметри заборонені."),0,true);
        else if( !owner().startStat( ) ) val.setS(_("2:Контролер
зупинено."),0,true);
        else val.setS("0",0,true);
    }
}

void TParamContr::setId( const string &vl )
{
    cfg("SHIFR").setS(vl);
}

void TParamContr::setType( const string &tpId )
{
    if(enableStat() || tpId == type().name ||
!owner().owner().tpPrmPresent(tpId)) return;

    setNodeMode(TCntrNode::Disable);

    try
    {
        //> Чекаємо поки роз'єднаються інші
        while(nodeUse(true) > 1) usleep(1000);
        //> Видаляємо з БД
        postDisable(true);

        //> Створюємо тимчасову структуру
        TConfig tCfg(&type());
        tCfg = *(TConfig*)this;

        //> Встановлюємо нову конфігурацію структури
        tipparm = &owner().owner().tpPrmAt(owner().owner().tpPrmToId(tpId));
        setElem(tipparm);

        //> Відновлюємо конфігурацію
        *(TConfig*)this = tCfg;
    }catch(...) { }
    setNodeMode(TCntrNode::Enable);
    setVlCfg(this);
    modif();
}

TVariant TParamContr::objFuncCall( const string &iid, vector<TVariant> &prms,
const string &user )
{
    //> Виклик функцій конфігурації
    TVariant cfRez = objFunc(iid, prms, user);
    if(!cfRez.isNull()) return cfRez;
    return TValue::objFuncCall(iid, prms, user);
}

void TParamContr::cntrCmdProc( XMLNode *opt )
{
    string a_path = opt->attr("path");
    //> Сервіс управління процесами
    if(a_path.substr(0,6) == "/serv/") { TValue::cntrCmdProc(opt); return; }
}

```

```

//> Беремо сторінку інформації
if(opt->name() == "info")
{
    TValue::cntrCmdProc(opt);
    ctrMkNode("Cisco_Application_Centric_Infrastructure_cntr",opt,-
1, "/", _("Параметр: ") + name(), RWRWR_, "root", SDAQ_ID);
    if(ctrMkNode("area", opt, 0, "/prm", _("Parameter")))
    {
        if(ctrMkNode("area", opt, -1, "/prm/st", _("Стан")))
        {
            if(!enableStat() && owner().owner().tpPrmSize() > 1)
                ctrMkNode("fld", opt, -
1, "/prm/st/type", _("Тип"), RWRWR_, "root", SDAQ_ID, 4, "tp", "str", "dest", "select", "select", "/prm/tpLst",
                "help", _("Змінюємо тип керівництва до останніх даних для
специфічних конфігурацій."));
            else ctrMkNode("fld", opt, -
1, "/prm/st/type", _("Тип"), R_R_R_, "root", SDAQ_ID, 1, "tp", "str");
            if(owner().enableStat())
                ctrMkNode("fld", opt, -
1, "/prm/st/en", _("Дозволено"), RWRWR_, "root", SDAQ_ID, 1, "tp", "bool");
        }
        if(ctrMkNode("area", opt, -1, "/prm/cfg", _("Configuration")))
            TConfig::cntrCmdMake(opt, "/prm/cfg", 0, "root", SDAQ_ID, RWRWR_);
    }
    return;
}
//> Процес управління на сторінці
if(a_path == "/prm/st/type")
{
    if(ctrChkNode(opt, "get", RWRWR_, "root", SDAQ_ID, SEC_RD)) opt->setText(type().name());
    if(ctrChkNode(opt, "set", RWRWR_, "root", SDAQ_ID, SEC_WR)) setType(opt->text());
}
else if(a_path == "/prm/st/en")
{
    if(ctrChkNode(opt, "get", RWRWR_, "root", SDAQ_ID, SEC_RD)) opt->setText(enableStat()?"1":"0");
    if(ctrChkNode(opt, "set", RWRWR_, "root", SDAQ_ID, SEC_WR))
    {
        if(!owner().enableStat()) throw TError(nodePath().c_str(), _("Контролер не запустився!"));
        else atoi(opt->text().c_str())?enable():disable();
    }
}
else if(a_path.substr(0,8) == "/prm/cfg")
    TConfig::cntrCmdProc(opt, TSYS::pathLev(a_path,2), "root", SDAQ_ID, RWRWR_);
else if(a_path == "/prm/templList" && ctrChkNode(opt))
{
    vector<string> lls, ls;
    SYS->daq().at().templLibList(lls);
    for(unsigned i_l = 0; i_l < lls.size(); i_l++)
    {
        SYS->daq().at().templLibAt(lls[i_l]).at().list(ls);
        for(unsigned i_t = 0; i_t < ls.size(); i_t++)
            opt->childAdd("el")->setText(lls[i_l]+"."+ls[i_t]);
    }
}
else if(a_path == "/prm/tpLst" && ctrChkNode(opt))
    for(unsigned i_tp = 0; i_tp < owner().owner().tpPrmSize(); i_tp++)
        opt->childAdd("el")->setAttr("id", owner().owner().tpPrmAt(i_tp).name)->setText(owner().owner().tpPrmAt(i_tp).descr);
    else TValue::cntrCmdProc(opt);
}

```

main.cpp - головна програма

```

//Файл системи керування мережною інфраструктурою на основі Cisco Application
Centric Infrastructure: main.cpp

#include <getopt.h>
#include "terror.h"
#include "tmess.h"
#include "tsys.h"

using namespace Cisco_Application_Centric_Infrastructure;

int main(int argc, char *argv[], char *envp[] )
{
    int rez = 0;

    //Перевірка початку роботи режиму основного процесу
    int next_opt;
    optind=opterr=0;
    struct option long_opt[] = { {"Режим основного процесу" ,0,NULL,'d'}, {NULL
,0,NULL,0 } };
    while((next_opt=getopt_long(argc,argv,"",long_opt,NULL)) != -1)
        if( next_opt == 'd' )
            {
                printf("Початок роботи режиму основного процесу!\n");
                int pid = fork();
                if( pid == -1 )
                    {
                        printf("Помилка: неможливо створити новий процес!\n");
                        return -1;
                    }
                if( pid != 0 )      return 0;

                //Готується оточення режиму основного процесу
                setsid();
                break;
            }

    try
    {
        SYS = new TSYS(argc,argv,envp);

        SYS->load();
        if( (rez=SYS->stopSignal()) > 0 ) return rez;
        rez = SYS->start();

        delete SYS;
    }catch(TError err) { mess_err(err.cat.c_str(),"%s",err.mess.c_str()); }

    printf("Система керування мережною інфраструктурою на основі Cisco
Application Centric Infrastructure коректно працює з даними %d.\n",rez);

    return rez;
}

```

```

//WebCisco_Application_Centric_Infrastructure системний файл: tsys.cpp

#include <features.h>
#include <ieee754.h>
#include <syscall.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <sys/stat.h>
#include <sys/time.h>
#include <sys/utsname.h>
#include <errno.h>
#include <string.h>
#include <fcntl.h>
#include <unistd.h>
#include <getopt.h>
#include <stdio.h>
#include <signal.h>
#include <stdarg.h>
#include <stdlib.h>
#include <langinfo.h>
#include <zlib.h>

#include "terror.h"
#include "tmess.h"
#include "tsys.h"

using namespace Cisco_Application_Centric_Infrastructure;

//Поточна зміна доступу
TMess *Cisco_Application_Centric_Infrastructure::Mess;
TSYS *Cisco_Application_Centric_Infrastructure::SYS;
bool TSYS::finalKill = false;
pthread_key_t TSYS::sTaskKey;

TSYS::TSYS( int argi, char ** argb, char **env ) : argc(argi), argv((const char
**)argb), envp((const char **)env),
    mUser("root"),
mConfFile("/etc/Cisco_Application_Centric_Infrastructure.xml"), mId("EmptySt"),
mName(_("Порожня Станція")), mIcoDir("./icons/"), mModDir("./"),
    mWorkDB("<cfg>"), mSaveAtExit(false), mSavePeriod(0), rootModifCnt(0),
mStopSignal(-1), mMultCPU(false)
{
    finalKill = false;
    SYS = this;          //Ініціалізуємо значення глобальних змінних доступу
    mSubst = grpAdd("sub_",true);
    nodeEn();
    pthread_key_create(&sTaskKey, NULL);

    Mess = new TMess();

    if(getenv("USER")) mUser = getenv("USER");

    //> Ініціалізуємо системний годинник
    clkCalc();

#ifdef __GLIBC_PREREQ(2,4)
    //> Multi CPU дозволяють перевірку
    cpu_set_t cpuset;
    CPU_ZERO(&cpuset);
    CPU_SET(1,&cpuset);
    mMultCPU = !pthread_setaffinity_np(pthread_self(), sizeof(cpu_set_t),
&cpuset);
#endif

    //> Встановлюємо сигнальні програми обробки
    signal(SIGINT,sighandler);

```

```

    signal(SIGTERM, sighandler);
    //signal(SIGCHLD, sighandler);
    signal(SIGALRM, sighandler);
    signal(SIGPIPE, sighandler);
    //signal(SIGFPE, sighandler);
    //signal(SIGSEGV, sighandler);
    signal(SIGABRT, sighandler);
}

TSYS::~TSYS( )
{
    finalKill = true;

    //Видаляємо всі вузли в команді управління
    del("ModSched");
    del("UI");
    del("Special");
    del("Archive");
    del("DAQ");
    del("Protocol");
    del("Transport");
    del("Security");
    del("BD");

    delete Mess;
    pthread_key_delete(sTaskKey);
}

string TSYS::host( )
{
    utsname ubuf; uname(&ubuf);
    return ubuf.nodename;
}

string TSYS::workDir( )
{
    char buf[STR_BUF_LEN];
    return getcwd(buf, sizeof(buf));
}

void TSYS::setWorkDir( const string &wdir )
{
    if(workDir() == wdir) return;
    if(chdir(wdir.c_str()) != 0)
        mess_warning(nodePath().c_str(), _("Змініть робочу директорію'%s' Помилка:
%s. Можливо поточний каталог вже встановлено правильно'%s'."),
        wdir.c_str(), strerror(errno), workDir().c_str());
    modif( );
}

XMLNode *TSYS::cfgNode( const string &path, bool create )
{
    string s_el, ndNm;

    XMLNode *t_node = &rootN;
    if(t_node->name() != "WebCisco_Application_Centric_Infrastructure")
    {
        if(!create) return NULL;
        t_node->setName("WebCisco_Application_Centric_Infrastructure");
    }

    for(int l_off = 0, nLev = 0; true; nLev++)
    {
        s_el = TSYS::pathLev(path, 0, true, &l_off);
        if(s_el.empty()) return t_node;
        bool ok = false;
        for(unsigned i_f = 0; !ok && i_f < t_node->childSize(); i_f++)
            if(t_node->childGet(i_f)->attr("id") == s_el)
            {

```

```

        t_node = t_node->childGet(i_f);
        ok = true;
    }
    if(!ok)
    {
        if(!create)    return NULL;
        ndNm = "prm";
        switch(nLev)
        {
            case 0: ndNm = "station";    break;
            case 1: if(s_el.compare(0,4,"sub_") == 0) ndNm = "node";    break;
            case 2: if(s_el.compare(0,4,"mod_") == 0) ndNm = "node";    break;
        }
        if(ndNm == "prm") t_node = t_node->childIns(0,ndNm)-
>setAttr("id",s_el);
        else t_node = t_node->childAdd(ndNm)->setAttr("id",s_el);
    }
    }
    return t_node;
}

string TSYS::int2str( int val, TSYS::IntView view )
{
    char buf[STR_BUF_LEN];
    if(view == TSYS::Dec)    snprintf(buf,sizeof(buf),"d",val);
    else if(view == TSYS::Oct)    snprintf(buf,sizeof(buf),"o",val);
    else if(view == TSYS::Hex)    snprintf(buf,sizeof(buf),"x",val);

    return buf;
}

string TSYS::uint2str( unsigned val, IntView view )
{
    char buf[STR_BUF_LEN];
    if(view == TSYS::Dec)    snprintf(buf,sizeof(buf),"u",val);
    else if(view == TSYS::Oct)    snprintf(buf,sizeof(buf),"o",val);
    else if(view == TSYS::Hex)    snprintf(buf,sizeof(buf),"x",val);

    return buf;
}

string TSYS::ll2str( int64_t val, IntView view )
{
    char buf[STR_BUF_LEN];
    if(view == TSYS::Dec)    snprintf(buf,sizeof(buf),"lld",val);
    else if(view == TSYS::Oct)    snprintf(buf,sizeof(buf),"llo",val);
    else if(view == TSYS::Hex)    snprintf(buf,sizeof(buf),"llx",val);

    return buf;
}

string TSYS::real2str( double val, int prec, char tp )
{
    char buf[STR_BUF_LEN];
    if(tp == 'g')    snprintf(buf,sizeof(buf),"%.*g",prec,val);
    else if(tp == 'e')    snprintf(buf,sizeof(buf),"%.*e",prec,val);
    else    snprintf(buf,sizeof(buf),"%.*f",prec,val);

    return buf;
}

string TSYS::time2str( time_t itm, const string &format )
{
    struct tm tm_tm;
    localtime_r(&itm,&tm_tm);
    char buf[100];
    int ret = strftime(buf, sizeof(buf), format.empty()?"%d-%m-%Y
%H:%M:%S":format.c_str(), &tm_tm);
    return (ret > 0) ? string(buf,ret) : string("");
}

```

```

}

string TSYS::time2str( double utm )
{
    if(utm < 1e-6) return "0";
    int lev = 0;
    int days = (int)floor(utm/(24*60*60*1e6));
    int часы = (int)floor(utm/(60*60*1e6))%24;
    int mins = (int)floor(utm/(60*1e6))%60;
    double usec = utm - 1e6*(days*24*60*60 + часы*60*60 + mins*60);

    string rez;
    if(days)          { rez += TSYS::int2str(days)+"day"; lev =
vmax(lev,6); }
    if(часы)          { rez += (rez.size()?"
":"" )+TSYS::int2str(часы)+"годин"; lev = vmax(lev,5); }
    if(mins && lev < 6) { rez += (rez.size()?"
":"" )+TSYS::int2str(mins)+"хвилини"; lev = vmax(lev,4); }
    if((1e-6*usec) > 0.5 && lev < 5)      { rez += (rez.size()?"
":"" )+TSYS::real2str(1e-6*usec,3)+"секунд"; lev = vmax(lev,3); }
    else if((1e-3*usec) > 0.5 && !lev)    { rez += (rez.size()?"
":"" )+TSYS::real2str(1e-3*usec,4)+"мікросекунд"; lev = vmax(lev,2); }
    else if(usec > 0.5 && !lev)          { rez += (rez.size()?"
":"" )+TSYS::real2str(usec,4)+"us"; lev = vmax(lev,1); }
    else if(!lev) rez += (rez.size()?" ":"") +TSYS::real2str(1e3*usec,4)+"ns";
    return rez;
}

string TSYS::cpct2str( double cnt )
{
    if(cnt > 0.2*pow(2,80)) return
TSYS::real2str(cnt/pow(2,80),3,'g')+"YiB";
    if(cnt > 0.2*pow(2,70)) return
TSYS::real2str(cnt/pow(2,70),3,'g')+"ZiB";
    if(cnt > 0.2*pow(2,60)) return
TSYS::real2str(cnt/pow(2,60),3,'g')+"EiB";
    if(cnt > 0.2*pow(2,50)) return
TSYS::real2str(cnt/pow(2,50),3,'g')+"PiB";
    if(cnt > 0.2*pow(2,40)) return
TSYS::real2str(cnt/pow(2,40),3,'g')+"TiB";
    if(cnt > 0.2*pow(2,30)) return
TSYS::real2str(cnt/pow(2,30),3,'g')+"GiB";
    if(cnt > 0.2*pow(2,20)) return
TSYS::real2str(cnt/pow(2,20),3,'g')+"MiB";
    if(cnt > 0.2*pow(2,10)) return
TSYS::real2str(cnt/pow(2,10),3,'g')+"KiB";
    return TSYS::real2str(cnt,3,'g')+"B";
}

string TSYS::addr2str( void *addr )
{
    char buf[sizeof(void*)*2+3];
    sprintf(buf,sizeof(buf),"%p",addr);

    return buf;
}

void *TSYS::str2addr( const string &str )
{
    return (void *)strtoul(str.c_str(),NULL,16);
}

string TSYS::strNoSpace( const string &val )
{
    int beg = -1, end = -1;

    for(unsigned i_s = 0; i_s < val.size(); i_s++)
        if(val[i_s] != ' ' && val[i_s] != '\n' && val[i_s] != '\t')
            {

```



```

//===== Редагування параметрів=====
int next_opt;
const char *short_opt="h";
struct option long_opt[] =
{
    {"help"      ,0,NULL,'h'},
    {"Config"   ,1,NULL,'f'},
    {"Station"  ,1,NULL,'s'},
    {NULL       ,0,NULL,0 }
};

optind=opterr=0;
do
{
    next_opt=getopt_long(argc, (char * const *)argv, short_opt, long_opt, NULL);
    switch(next_opt)
    {
        case 'h':
            fprintf(stdout, "%s", optDescr().c_str());
            Mess->setMessLevel(7);
            cmd_help = true;
            break;
        case 'f': mConfFile = optarg; break;
        case 's': mId = optarg; break;
        case -1 : break;
    }
} while(next_opt != -1);

//Завантажуємо конфігураційний файл
int hd = open(mConfFile.c_str(), O_RDONLY);
if(hd < 0) mess_err(nodePath().c_str(), _("Конфігураційний файл '%s' Помилка:
%s"), mConfFile.c_str(), strerror(errno));
else
{
    string s_buf;
    int cf_sz = lseek(hd, 0, SEEK_END);
    if(cf_sz > 0)
    {
        lseek(hd, 0, SEEK_SET);
        char *buf = (char *)malloc(cf_sz+1);
        read(hd, buf, cf_sz);
        buf[cf_sz] = 0;
        s_buf = buf;
        free(buf);
    }
    close(hd);

    try
    {
        ResAlloc res(nodeRes(), true);
        rootN.load(s_buf, true);
        if(rootN.name() == "WebCisco_Application_Centric_Infrastructure")
        {
            XMLNode *stat_n = NULL;
            for(int i_st = rootN.childSize()-1; i_st >= 0; i_st--)
                if(rootN.childGet(i_st)->name() == "station")
                {
                    stat_n = rootN.childGet(i_st);
                    if(stat_n->attr("id") == mId) break;
                }
            if(stat_n && stat_n->attr("id") != mId)
            {
                mess_warning(nodePath().c_str(), _("Робоча станція '%s' не
представлена у конфігураційному файлі Використайте '%s' конфігурацію робочої
станції!"),
                    mId.c_str(), stat_n->attr("id").c_str());
                mId = stat_n->attr("id");
            }
        }
    }
}

```

```

        if(!stat_n) rootN.clear();
    } else rootN.clear();
    if(!rootN.childSize()) mess_err(nodePath().c_str(),_("Помилка
конфігурації '%s'!"),mConfFile.c_str());
    rootModifCnt = 0;
}
catch(TError err) { mess_err(nodePath().c_str(),_("Завантажуємо
конфігураційний файл Помилка: %s"),err.mess.c_str() ); }
}

return cmd_help;
}

void TSYS::cfgFileSave( )
{
    ResAlloc res(nodeRes(),true);
    if(!rootModifCnt) return;
    int hd = open(mConfFile.c_str(), O_CREAT|O_TRUNC|O_WRONLY, 0664);
    if(hd < 0) mess_err(nodePath().c_str(),_("Конфігураційний файл '%s' Помилка:
%s"),mConfFile.c_str(),strerror(errno));

    string rezFile = rootN.save(XMLNode::XMLHeader);
    int rez = write(hd, rezFile.data(), rezFile.size());
    if(rez != (int)rezFile.size()) mess_err(nodePath().c_str(),_("Помилка запису
конфігурації. %s"),mConfFile.c_str(),((rez<0)?strerror(errno):""));
    rootModifCnt = 0;
    rootFlTm = time(NULL);
}

void TSYS::cfgPrmLoad( )
{
    //Системні параметри
    mName =
TBDS::genDBGet (nodePath()+"StName",name(),"root",TBDS::UseTranslate);
mWorkDB = TBDS::genDBGet (nodePath()+"WorkDB",workDB(),"root",TBDS::OnlyCfg);
setWorkDir (TBDS::genDBGet (nodePath()+"Workdir").c_str());
setIcoDir (TBDS::genDBGet (nodePath()+"IcoDir",icoDir()));
setModDir (TBDS::genDBGet (nodePath()+"ModDir",modDir()));
setSaveAtExit (atoi (TBDS::genDBGet (nodePath()+"SaveAtExit","0").c_str()));
setSavePeriod (atoi (TBDS::genDBGet (nodePath()+"SavePeriod","0").c_str()));
}

void TSYS::load_()
{
    static bool first_load = true;

    bool cmd_help = cfgFileLoad();
    mess_info (nodePath().c_str(),_("Load!"));
    cfgPrmLoad();
    Mess->load(); //Завантажуємо повідомлення

    if( first_load )
    {
        //> Створюємо підсистему
        add( new TBDS() );
        add( new TSecurity() );
        add( new TTransportS() );
        add( new TProtocols() );
        add( new TDAQS() );
        add( new TArchives() );
        add( new TSpecialS() );
        add( new TUIS() );
        add( new TModSchedul() );

        //> Завантажуємо модулі
        modSchedul().at().load();
        if( !modSchedul().at().loadLibS() )
        {

```

```

        mess_err(nodePath().c_str(),_("Жоден модуль не завантажений. Ваша
конфігурація перервана!"));
        stop();
    }

    //> Завантажуємо базу даних першої підсистеми
    db().at().load();
    if( !cmd_help ) modSchedul().at().modifG();    // Для перевантаження
спроби від бази даних

    //> Друге завантаження для завантаження від родової БД
    Mess->load();
    cfgPrmLoad();
}

//> Пряме завантаження підсистем та модулів
vector<string> lst;
list(lst);
for( unsigned i_a=0; i_a < lst.size(); i_a++ )
    try { at(lst[i_a]).at().load(); }
    catch(TError err)
    {
        mess_err(err.cat.c_str(), "%s", err.mess.c_str());
        mess_err(nodePath().c_str(),_("Помилка завантаження підсистеми
'%s'. "),lst[i_a].c_str());
    }

    if( cmd_help ) stop();
    first_load = false;
}

void TSYS::save_( )
{
    char buf[STR_BUF_LEN];

    mess_info(nodePath().c_str(),_("Save!"));

    //> Системні параметри
    getcwd(buf, sizeof(buf));
    TBDS::genDBSet(nodePath()+"StName", mName, "root", TBDS::UseTranslate);
    TBDS::genDBSet(nodePath()+"Workdir", buf);
    TBDS::genDBSet(nodePath()+"IcoDir", icoDir());
    TBDS::genDBSet(nodePath()+"ModDir", modDir());
    TBDS::genDBSet(nodePath()+"SaveAtExit", TSYS::int2str(saveAtExit()));
    TBDS::genDBSet(nodePath()+"SavePeriod", TSYS::int2str(savePeriod()));

    Mess->save(); //Завантажуємо повідомлення
}

int TSYS::start( )
{
    vector<string> lst;
    list(lst);

    mess_info(nodePath().c_str(),_("Start!"));
    for(unsigned i_a=0; i_a < lst.size(); i_a++)
        try { at(lst[i_a]).at().subStart(); }
        catch(TError err)
        {
            mess_err(err.cat.c_str(), "%s", err.mess.c_str());
            mess_err(nodePath().c_str(),_("Помилка запуску
підсистеми '%s'. "),lst[i_a].c_str());
        }

    cfgFileScan( true );

    mess_info(nodePath().c_str(),_("Завершення запуску!"));

    unsigned int i_cnt = 1;

```

```

mStopSignal = 0;
while(!mStopSignal)
{
    //> CPU підрахунок частоти
    if(!(i_cnt%(10*1000/STD_WAIT_DELAY)))    clkCalc( );

    //> Кофігураційний файл змін періодичних перевірок
    if(!(i_cnt%(10*1000/STD_WAIT_DELAY)))    cfgFileScan( );

    //> Періодична загальна перевірка бібліотек
    if(modSchedul( ).at( ).chkPer( ) && !(i_cnt%(modSchedul(
).at( ).chkPer( )*1000/STD_WAIT_DELAY)))
        modSchedul( ).at( ).libLoad(modDir( ),true);

    //> Періодичний запис змін до БД
    if(savePeriod( ) && !(i_cnt%(savePeriod( )*1000/STD_WAIT_DELAY))) save( );

    //> Кофігураційний файл зберігає необхідні зміни
    if(!(i_cnt%(10*1000/STD_WAIT_DELAY)))    cfgFileSave( );

    //> Викликаємо підсистему кожні 10 сек.
    if(!(i_cnt%(10*1000/STD_WAIT_DELAY)))
        for(unsigned i_a=0; i_a < lst.size(); i_a++)
            try { at(lst[i_a]).at( ).perSYSCall(i_cnt/(1000/STD_WAIT_DELAY)); }
            catch(TError err) { mess_err(err.cat.c_str( ),"%s",err.mess.c_str( )); }
}

    usleep(STD_WAIT_DELAY*1000);
    i_cnt++;
}

mess_info(nodePath( ).c_str( ),_("Stop!"));
if(saveAtExit( ) || savePeriod( ))    save( );
cfgFileSave( );
for(int i_a=lst.size()-1; i_a >= 0; i_a--)
    try { at(lst[i_a]).at( ).subStop( ); }
    catch(TError err)
    {
        mess_err(err.cat.c_str( ),"%s",err.mess.c_str( ));
        mess_err(nodePath( ).c_str( ),_("Помилка остановки
підсистеми'%s'."),lst[i_a].c_str( ));
    }

    return mStopSignal;
}

void TSYS::stop( )
{
    mStopSignal = SIGUSR1;
}

bool TSYS::chkSelDB( const string& wDB, bool isStrong )
{
    if(selDB( ).empty( ) && !isStrong) return true;
    if(SYS->selDB( ) == TBDS::realDBName(wDB)) return true;
    return false;
}

void TSYS::sighandler( int signal )
{
    switch(signal)
    {
        case SIGINT:
            SYS->mStopSignal=signal;
            break;
        case SIGTERM:
            mess_warning(SYS->nodePath( ).c_str( ),_("Отриманий сигнал переривання
роботи. Сервер зупиняється!"));
            SYS->mStopSignal=signal;
    }
}

```

```

        break;
    case SIGFPE:
        mess_warning(SYS->nodePath().c_str(),_("Виключення плаваючої крапки
спіймане!"));
        exit(1);
        break;
    case SIGCHLD:
    {
        int status;
        pid_t pid = wait(&status);
        if(!WIFEXITED(status) && pid > 0)
            mess_info(SYS->nodePath().c_str(),_("Вивільнено процес-
потомок%d!"),pid);
        break;
    }
    case SIGPIPE:
        //mess_warning(SYS->nodePath().c_str(),_("Сигнал переривання
PIPE!"));
        break;
    case SIGSEGV:
        mess_emerg(SYS->nodePath().c_str(),_("Сигнал помилки від сегменту!"));
        break;
    case SIGABRT:
        mess_emerg(SYS-
>nodePath().c_str(),_("WebCisco_Application_Centric_Infrastructure перервала
роботу!"));
        break;
    case SIGALRM:      break;
    default:
        mess_warning(SYS->nodePath().c_str(),_("Невизначений
сигнал%d!"),signal);
    }
}

void TSYS::cfgFileScan( bool first )
{
    struct stat f_stat;

    if(stat(cfgFile().c_str(),&f_stat) != 0) return;
    bool up = false;
    if(rootCfgFl != cfgFile() || rootFlTm != f_stat.st_mtime) up = true;
    rootCfgFl = cfgFile();
    rootFlTm = f_stat.st_mtime;

    if(up && !first)
    {
        modifG();
        setSelDB("<cfg>");
        load();
        setSelDB("");
    }
}

int64_t TSYS::curTime( )
{
    timeval cur_tm;
    gettimeofday(&cur_tm,NULL);
    return (int64_t)cur_tm.tv_sec*1000000 + cur_tm.tv_usec;
}

bool TSYS::eventWait( bool &m_mess_r_stat, bool exempl, const string &loc,
time_t tm )
{
    time_t t_tm, s_tm;

    t_tm = s_tm = time(NULL);
    while( m_mess_r_stat != exempl )
    {
        time_t c_tm = time(NULL);

```

```

//Контролюємо перерву
if( tm && ( c_tm > s_tm+tm )
{
    mess_crit(loc.c_str(),_("Timeouted !!!"));
    return true;
}
//Створюємо повідомлення
if( c_tm > t_tm+1 ) //1sec
{
    t_tm = c_tm;
    mess_info(loc.c_str(),_("Чекаємо подію..."));
}
usleep(STD_WAIT_DELAY*1000);
}
return false;
}

string TSYS::strSepParse( const string &path, int level, char sep, int *off )
{
    int an_dir = off ? *off : 0;
    int t_lev = 0;
    size_t t_dir;

    if(an_dir >= (int)path.size()) return "";
    while(true)
    {
        t_dir = path.find(sep,an_dir);
        if( t_dir == string::npos )
        {
            if( off ) *off = path.size();
            return (t_lev == level) ? path.substr(an_dir) : "";
        }
        else if( t_lev == level )
        {
            if( off ) *off = t_dir+1;
            return path.substr(an_dir,t_dir-an_dir);
        }
        an_dir = t_dir+1;
        t_lev++;
    }
    return "";
}

string TSYS::strParse( const string &path, int level, const string &sep, int
*off, bool mergeSepSymb )
{
    int an_dir = off ? *off : 0;
    int t_lev = 0;
    size_t t_dir;

    if(an_dir >= (int)path.size() || sep.empty()) return "";
    while(true)
    {
        t_dir = path.find(sep,an_dir);
        if( t_dir == string::npos )
        {
            if( off ) *off = path.size();
            return (t_lev == level) ? path.substr(an_dir) : "";
        }
        else if( t_lev == level )
        {
            if( off ) *off = t_dir+sep.size();
            return path.substr(an_dir,t_dir-an_dir);
        }
        if( mergeSepSymb && sep.size() == 1 )
            for(an_dir = t_dir; an_dir < (int)path.size() && path[an_dir] ==
sep[0]; ) an_dir++;
        else an_dir = t_dir+sep.size();
        t_lev++;
    }
}

```

```

    }
    return "";
}

string TSYS::strLine( const string &str, int level, int *off )
{
    int an_dir = off ? *off : 0;
    int t_lev = 0, edLnSmbSz = 1;
    size_t t_dir;

    if(an_dir >= (int)str.size()) return "";
    while(true)
    {
        for(t_dir = an_dir; t_dir < str.size(); t_dir++)
            if(str[t_dir] == '\x0D' || str[t_dir] == '\x0A')
                { edLnSmbSz = (str[t_dir] == '\x0D' && ((t_dir+1) < str.size()) &&
str[t_dir+1] == '\x0A') ? 2 : 1; break; }
            if(t_dir >= str.size())
                {
                    if(off) *off = str.size();
                    return (t_lev==level) ? str.substr(an_dir) : "";
                }
            else if(t_lev == level)
                {
                    if(off) *off = t_dir+edLnSmbSz;
                    return str.substr(an_dir,t_dir-an_dir);
                }
            an_dir = t_dir+edLnSmbSz;
            t_lev++;
        }
    return "";
}

string TSYS::pathLev( const string &path, int level, bool encode, int *off )
{
    int an_dir = off ? *off : 0;
    int t_lev = 0;
    size_t t_dir;

    //> Перший роздільний прохід
    while(an_dir < (int)path.size() && path[an_dir]=='/') an_dir++;
    if(an_dir >= (int)path.size()) return "";
    //> Шлях рівня процесу
    while(true)
    {
        t_dir = path.find("/",an_dir);
        if( t_dir == string::npos )
            {
                if( off ) *off = path.size();
                return (t_lev == level) ? ( encode ?
TSYS::strDecode(path.substr(an_dir),TSYS::PathEl) : path.substr(an_dir) ) : "";
            }
        else if( t_lev == level )
            {
                if( off ) *off = t_dir;
                return encode ? TSYS::strDecode(path.substr(an_dir,t_dir-
an_dir),TSYS::PathEl) : path.substr(an_dir,t_dir-an_dir);
            }
        an_dir = t_dir;
        t_lev++;
        while(an_dir < (int)path.size() && path[an_dir]=='/') an_dir++;
    }
}

string TSYS::path2sepstr( const string &path, char sep )
{
    string rez, curv;
    int off = 0;
    while( !(curv=TSYS::pathLev(path,0,false,&off)).empty() )

```

```

        rez+=curv+sep;
        if(!rez.empty())    rez.resize(rez.size()-1);

        return rez;
    }

string TSYS::sepstr2path( const string &str, char sep )
{
    string rez, curv;
    int off = 0;
    while( !(curv=TSYS::strSepParse(str,0,sep,&off)).empty() )
        rez+="/"+curv;

    return rez;
}

string TSYS::strEncode( const string &in, TSYS::Code tp, const string &symb )
{
    int i_sz;
    string sout;

    switch(tp)
    {
        case TSYS::PathEl:
            sout = in;
            for( i_sz = 0; i_sz < (int)sout.size(); i_sz++ )
                switch( sout[i_sz] )
                {
                    case '/': sout.replace(i_sz,1,"%2f"); i_sz+=2; break;
                    case '%': sout.replace(i_sz,1,"%25"); i_sz+=2; break;
                }
            break;
        case TSYS::HttpURL:
            sout = in;
            for( i_sz = 0; i_sz < (int)sout.size(); i_sz++ )
                switch( sout[i_sz] )
                {
                    case '%': sout.replace(i_sz,1,"%25"); i_sz+=2; break;
                    case ' ': sout.replace(i_sz,1,"%20"); i_sz+=2; break;
                    case '\t': sout.replace(i_sz,1,"%09"); i_sz+=2; break;
                    default:
                        if( sout[i_sz]&0x80 )
                        {
                            char buf[4];
                            sprintf(buf,sizeof(buf),"%02X", (unsigned
char)sout[i_sz]);
                            sout.replace(i_sz,1,buf);
                            i_sz+=2;
                            break;
                        }
                }
            break;
        case TSYS::Html:
            sout.reserve(in.size()+10);
            for( i_sz = 0; i_sz < (int)in.size(); i_sz++ )
                switch( in[i_sz] )
                {
                    case '>':    sout+="&gt;";    break;
                    case '<':    sout+="&lt;";    break;
                    case '"':    sout+="&quot;";  break;
                    case '&':    sout+="&amp;";  break;
                    case '\':    sout+="&apos;";  break;
                    default:     sout+=in[i_sz];
                }
            break;
        case TSYS::JavaSc:
            sout.reserve(in.size()+10);
            for( i_sz = 0; i_sz < (int)in.size(); i_sz++ )
                switch( in[i_sz] )

```

```

        {
            case '\n':    sout+="\\n";    break;
            default:    sout+=in[i_sz];
        }
    }
    break;
case TSYS::SQL:
    sout.reserve(in.size()+10);
    for( i_sz = 0; i_sz < (int)in.size(); i_sz++ )
        switch( in[i_sz] )
        {
            case '\\':    sout+="\\";    break;
            case '\\"':   sout+="\\";    break;
            case '`':     sout+="`";    break;
            case '\\\'':  sout+="\\'";  break;
            default:     sout+=in[i_sz];
        }
    }
    break;
case TSYS::Custom:
    sout.reserve(in.size()+10);
    for( i_sz = 0; i_sz < (int)in.size(); i_sz++ )
    {
        unsigned i_smb;
        for(i_smb = 0; i_smb < symb.size(); i_smb++)
            if(in[i_sz] == symb[i_smb])
            {
                char buf[4];
                sprintf(buf,"%02X", (unsigned char)in[i_sz]);
                sout += buf;
                break;
            }
        if(i_smb >= symb.size()) sout += in[i_sz];
    }
    break;
case TSYS::base64:
    {
        sout.reserve(in.size()+in.size()/4+in.size()/57+10);
        const char *base64alph =
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/";
        for( i_sz = 0; i_sz < (int)in.size(); i_sz+=3 )
        {
            if(i_sz && !(i_sz%57)) sout.push_back('\n');
            sout.push_back(base64alph[(unsigned char)in[i_sz]>>2]);
            if((i_sz+1) >= (int)in.size())
            {
                sout.push_back(base64alph[((unsigned char)in[i_sz]&0x03)<<4]);
                sout += "==";
            }
            else
            {
                sout.push_back(base64alph[((unsigned
char)in[i_sz]&0x03)<<4]|((unsigned char)in[i_sz+1]>>4)]);
                if((i_sz+2) >= (int)in.size())
                {
                    sout.push_back(base64alph[((unsigned
char)in[i_sz+1]&0x0F)<<2]);
                    sout.push_back('=');
                }
                else
                {
                    sout.push_back(base64alph[((unsigned
char)in[i_sz+1]&0x0F)<<2]|((unsigned char)in[i_sz+2]>>6)]);
                    sout.push_back(base64alph[(unsigned char)in[i_sz+2]&0x3F]);
                }
            }
        }
    }
    break;
}
case TSYS::FormatPrint:
    sout = in;

```

```

for(i_sz = 0; i_sz < (int)sout.size(); i_sz++)
    if(sout[i_sz] == '%') { sout.replace(i_sz,1,"%%"); i_sz++; }
break;
case TSYS::oscdID:
sout.reserve(in.size());
for(i_sz = 0; i_sz < (int)in.size(); i_sz++)
    switch(in[i_sz])
    {
        case ' ': case '/': case '\\': case '&': case '(':
        case ')': case '[': case ']': case '!': case '~':
        case '`': case '@': case '%': case '^': case '-':
        case '+': case '=': case '*': case '{': case '}':
        case ':': case ';': case '"': case '\\': case '<':
        case '>': case '?': case '.': case ',':
            sout+="_"; break;
        default:      sout+=in[i_sz];
    }
break;
case TSYS::Bin:
{
    string svl, evl;
sout.reserve(in.size());
for(int off = 0; (svl=TSYS::strSepParse(in,0,'\n',&off)).size(); )
    for(int offE = 0; (evl=TSYS::strSepParse(svl,0,' ',&offE)).size(); )
        sout+=(char)strtol(evl.c_str(),NULL,16);
break;
}
case TSYS::Reverse:
    for(i_sz = in.size()-1; i_sz >= 0; i_sz--) sout += in[i_sz];
break;
case TSYS::ShieldSimb:
sout.reserve(in.size());
for(i_sz = 0; i_sz < (int)in.size(); i_sz++)
    if(in[i_sz] == '\\') && i_sz < ((int)in.size()-1)
    {
        switch(in[i_sz+1])
        {
            case 'a':  sout += '\\a';    break;
            case 'b':  sout += '\\b';    break;
            case 'f':  sout += '\\f';    break;
            case 'n':  sout += '\\n';    break;
            case 'r':  sout += '\\r';    break;
            case 't':  sout += '\\t';    break;
            case 'v':  sout += '\\v';    break;
            case 'x':  case 'X':
                if((i_sz+3) < (int)in.size() && isxdigit(in[i_sz+2]) &&
isxdigit(in[i_sz+3]))
                    { sout +=
(char)strtol(in.substr(i_sz+2,2).c_str(),NULL,16); i_sz += 2; }
                    else sout += in[i_sz+1];
                    break;
            default:
                if((i_sz+3) < (int)in.size() && in[i_sz+1] >= '0' &&
in[i_sz+1] <= '7' &&
in[i_sz+2] >= '0' &&
in[i_sz+2] <= '7' &&
in[i_sz+3] >= '0' &&
in[i_sz+3] <= '7')
                    { sout +=
(char)strtol(in.substr(i_sz+1,3).c_str(),NULL,8); i_sz += 2; }
                    else sout += in[i_sz+1];
                }
            i_sz++;
        }else sout += in[i_sz];
break;
}
return sout;
}

```

```

unsigned char TSYs::getBase64Code(unsigned char asymb)
{
    switch(asymb)
    {
        case 'A' ... 'Z': return asymb-(unsigned char)'A';
        case 'a' ... 'z': return 26+asymb-(unsigned char)'a';
        case '0' ... '9': return 52+asymb-(unsigned char)'0';
        case '+':         return 62;
        case '/':         return 63;
    }
    return 0;
}

string TSYs::strDecode( const string &in, TSYs::Code tp )
{
    unsigned i_sz;
    string sout;

    switch(tp)
    {
        case TSYs::PathEl: case TSYs::HttpURL: case TSYs::Custom:
            sout.reserve(in.size());
            for(i_sz = 0; i_sz < in.size(); i_sz++)
                switch(in[i_sz])
                {
                    case '%':
                        if(i_sz+2 < in.size())
                        {
                            sout += (char)strtol(in.substr(i_sz+1,2).c_str(),NULL,16);
                            i_sz += 2;
                        }else sout += in[i_sz];
                        break;
                    default: sout += in[i_sz];
                }
            break;
        case TSYs::base64:
            sout.reserve(in.size());
            for( i_sz = 0; i_sz < in.size(); )
            {
                if(in[i_sz] == '\n')    i_sz+=sizeof('\n');
                if((i_sz+3) < in.size())
                    if( in[i_sz+1] != '=' )
                    {
                        char w_code1 = TSYs::getBase64Code(in[i_sz+1]);

                        sout.push_back((TSYs::getBase64Code(in[i_sz])<<2) | (w_code1>>4));
                        if( in[i_sz+2] != '=' )
                        {
                            char w_code2 = TSYs::getBase64Code(in[i_sz+2]);
                            sout.push_back((w_code1<<4) | (w_code2>>2));
                            if( in[i_sz+3] != '=' )

                                sout.push_back((w_code2<<6) | TSYs::getBase64Code(in[i_sz+3]));
                        }
                    }
                i_sz+=4;
            }
            break;
        case TSYs::Bin:
            sout.reserve(in.size());
            for( i_sz = 0; i_sz < in.size(); i_sz++ )
                sout += TSYs::strMess(((i_sz+1)%16)?"%0.2x ":"%0.2x\n", (unsigned
char)in[i_sz]);
            break;
        default: sout = in;    break;
    }

    return sout;
}

```

```

string TSYS::strCompr( const string &in, int lev )
{
    z_stream strm;

    if( in.empty() )    return "";

    strm.zalloc = Z_NULL;
    strm.zfree  = Z_NULL;
    strm.opaque = Z_NULL;

    if( deflateInit(&strm,lev) != Z_OK ) return "";

    uLongf comprLen = deflateBound(&strm,in.size());
    char out[comprLen];

    strm.next_in  = (Bytef*)in.data();
    strm.avail_in = (uInt)in.size();
    strm.next_out = (Bytef*)out;
    strm.avail_out = comprLen;

    if( deflate(&strm, Z_FINISH) != Z_STREAM_END )
    {
        deflateEnd(&strm);
        return "";
    }

    comprLen = strm.total_out;

    deflateEnd(&strm);

    return string(out,comprLen);
}

string TSYS::strUncompr( const string &in )
{
    int ret;
    z_stream strm;
    unsigned char out[STR_BUF_LEN];
    string rez;

    if( in.empty() )    return "";

    strm.zalloc = Z_NULL;
    strm.zfree  = Z_NULL;
    strm.opaque = Z_NULL;

    if( inflateInit(&strm) != Z_OK )    return "";

    strm.avail_in = in.size();
    strm.next_in  = (Bytef*)in.data();
    do
    {
        {
            strm.avail_out = sizeof(out);
            strm.next_out = out;
            ret=inflate(&strm,Z_NO_FLUSH);
            if( ret == Z_STREAM_ERROR || ret == Z_NEED_DICT || ret == Z_DATA_ERROR ||
ret == Z_MEM_ERROR )
                break;
            rez.append((char*)out,sizeof(out)-strm.avail_out);
        } while( strm.avail_out == 0 );

        inflateEnd(&strm);

        if( ret != Z_STREAM_END ) return "";

        return rez;
    }
}

```

```

float TSYS::floatLE(float in)
{
#ifdef __BYTE_ORDER == __BIG_ENDIAN
    ieee754_double ieee754_be;
    union ieee754_le
    {
        float f;
        struct
        {
            unsigned int mantissa:23;
            unsigned int exponent:8;
            unsigned int negative:1;
        } ieee;
    } ieee754_le;

    ieee754_be.f = in;
    ieee754_le.ieee.mantissa = ieee754_be.ieee.mantissa;
    ieee754_le.ieee.exponent = ieee754_be.ieee.exponent;
    ieee754_le.ieee.negative = ieee754_be.ieee.negative;

    return ieee754_le.f;
#endif

    return in;
}

float TSYS::floatLErev(float in)
{
#ifdef __BYTE_ORDER == __BIG_ENDIAN
    ieee754_double ieee754_be;
    union ieee754_le
    {
        float f;
        struct
        {
            unsigned int mantissa:23;
            unsigned int exponent:8;
            unsigned int negative:1;
        } ieee;
    } ieee754_le;

    ieee754_le.f = in;
    ieee754_be.ieee.mantissa = ieee754_le.ieee.mantissa;
    ieee754_be.ieee.exponent = ieee754_le.ieee.exponent;
    ieee754_be.ieee.negative = ieee754_le.ieee.negative;

    return ieee754_be.f;
#endif

    return in;
}

double TSYS::doubleLE(double in)
{
#ifdef __BYTE_ORDER == __BIG_ENDIAN || __FLOAT_WORD_ORDER == __BIG_ENDIAN
    ieee754_double ieee754_be;
    union ieee754_le
    {
        double d;
        struct
        {
            unsigned int mantissa1:32;
            unsigned int mantissa0:20;
            unsigned int exponent:11;
            unsigned int negative:1;
        } ieee;
    } ieee754_le;

    ieee754_be.d = in;

```

```

    ieee754_le.ieee.mantissa0 = ieee754_be.ieee.mantissa0;
    ieee754_le.ieee.mantissa1 = ieee754_be.ieee.mantissa1;
    ieee754_le.ieee.exponent = ieee754_be.ieee.exponent;
    ieee754_le.ieee.negative = ieee754_be.ieee.negative;

    return ieee754_le.d;
#endif

    return in;
}

double TSYS::doubleLErev(double in)
{
#ifdef __BYTE_ORDER == __BIG_ENDIAN || __FLOAT_WORD_ORDER == __BIG_ENDIAN
    ieee754_double ieee754_be;
    union ieee754_le
    {
        double d;
        struct
        {
            unsigned int mantissa1:32;
            unsigned int mantissa0:20;
            unsigned int exponent:11;
            unsigned int negative:1;
        } ieee;
    } ieee754_le;

    ieee754_le.d = in;
    ieee754_be.ieee.mantissa0 = ieee754_le.ieee.mantissa0;
    ieee754_be.ieee.mantissa1 = ieee754_le.ieee.mantissa1;
    ieee754_be.ieee.exponent = ieee754_le.ieee.exponent;
    ieee754_be.ieee.negative = ieee754_le.ieee.negative;

    return ieee754_be.d;
#endif

    return in;
}

long TSYS::HZ ()
{
    return sysconf(_SC_CLK_TCK);
}

bool TSYS::cntrEmpty( )
{
    ResAlloc res( nodeRes(), false );
    return mCntrs.empty();
}

double TSYS::cntrGet( const string &id )
{
    ResAlloc res( nodeRes(), false );
    map<string,double>::iterator icnt = mCntrs.find(id);
    if( icnt == mCntrs.end() ) return 0;
    return icnt->second;
}

void TSYS::cntrSet( const string &id, double vl )
{
    ResAlloc res( nodeRes(), true );
    mCntrs[id] = vl;
}

void TSYS::taskCreate( const string &path, int priority, void
*( *start_routine)( void * ), void *arg, int wtm, pthread_attr_t *pAttr, bool
*startSt )
{
    int detachStat = 0;

```

```

pthread_t procPthr;
pthread_attr_t locPAttr, *pthr_attr;
map<string,STask>::iterator ti;

ResAlloc res(taskRes, true);
for(time_t c_tm = time(NULL); mTasks.find(path) != mTasks.end(); )
{
    if(time(NULL) >= (c_tm+wtm)) throw TError(nodePath().c_str(),_("Завдання
'%s' вже присутнє!"),path.c_str());
    res.release();
    usleep(10000);
    res.request(true);
}
STask &htsk = mTasks[path];
htsk.path = path;
htsk.task = start_routine;
htsk.taskArg = arg;
htsk.flgs = 0;
res.release();

if(pAttr) pthr_attr = pAttr;
else
{
    pthr_attr = &locPAttr;
    pthread_attr_init(pthr_attr);
}
pthread_attr_setinheritsched(pthr_attr, PTHREAD_EXPLICIT_SCHED);
struct sched_param prior;
prior.sched_priority = 0;

int policy = SCHED_OTHER;
#if __GLIBC_PREREQ(2,4)
if(priority < 0)    policy = SCHED_BATCH;
#endif
if(priority > 0 /*&& SYS->user() == "root"*/)    policy = SCHED_RR;
pthread_attr_setschedpolicy(pthr_attr, policy);
prior.sched_priority =
vmax(sched_get_priority_min(policy),vmin(sched_get_priority_max(policy),priority
));
pthread_attr_setschedparam(pthr_attr,&prior);

try
{
    pthread_attr_getdetachstate(pthr_attr,&detachStat);
    if(detachStat == PTHREAD_CREATE_DETACHED) htsk.flgs |= STask::Detached;
    int rez = pthread_create(&procPthr, pthr_attr, taskWrap, &htsk);
    if(rez == EPERM)
    {
        mess_warning(nodePath().c_str(),_("No permission for create real-time
policy. Default thread is created!"));
        policy = SCHED_OTHER;
        pthread_attr_setschedpolicy(pthr_attr, policy);
        prior.sched_priority = 0;
        pthread_attr_setschedparam(pthr_attr,&prior);
        rez = pthread_create(&procPthr, pthr_attr, taskWrap, &htsk);
    }
    if(!pAttr) pthread_attr_destroy(pthr_attr);

    if(rez) throw TError(nodePath().c_str(),_("Завдання створило
помилку%d."), rez);

    //> Чекаємо закінчення ініціалізації структури потоку для не відривних
завдань
    while(!(htsk.flgs&STask::Detached) && !htsk.thr) pthread_yield();
    //> Чекаємо запуск статусу
    for(time_t c_tm = time(NULL); !(htsk.flgs&STask::Detached) && startSt &&
!(*startSt); )
    {

```

```

        if(time(NULL) >= (c_tm+wtm)) throw
TError(nodePath().c_str(),_("Завдання '%s' запуск відкладений!"),path.c_str());
        usleep(STD_WAIT_DELAY *1000);
    }
}
catch(TError)
{
    res.request(true);
    mTasks.erase(path);
    res.release();
    throw;
}
}

void TSYS::taskDestroy( const string &path, bool *endrunCntr, int wtm, bool
noSignal )
{
    ResAlloc res(taskRes, false);
    map<string,STask>::iterator it = mTasks.find(path);
    if(it == mTasks.end()) return;
    pthread_t thr = it->second.thr;
    res.release();

    if(endrunCntr) *endrunCntr = true;
    if(!noSignal) pthread_kill(thr, SIGALRM);

    //> Чекаємо завершення завдання та повторюємо відправлення SIGALRM
    time_t t_tm, s_tm;
    t_tm = s_tm = time(NULL);
    while(!(it->second.flgs&STask::FinishTask))
    {
        if(!noSignal) pthread_kill(thr, SIGALRM);
        time_t c_tm = time(NULL);
        //Контролюємо перерву
        if(wtm && (c_tm > (s_tm+wtm)))
        {
            mess_crit((nodePath()+path+": stop").c_str(),_("Timeouted !!!"));
            throw TError(nodePath().c_str(),_("«завдання '%s' is not
stopped!"),path.c_str());
        }
        //Створюємо повідомлення
        if(c_tm > t_tm+1) //1sec
        {
            t_tm = c_tm;
            mess_info((nodePath()+path+": stop").c_str(),_("Чекаємо подію..."));
        }
        usleep(STD_WAIT_DELAY*1000);
    }

    if(!(it->second.flgs&STask::Detached)) pthread_join(thr, NULL);

    res.request(true);
    mTasks.erase(it);
}

void *TSYS::taskWrap( void *stas )
{
    //> Беремо тимчасову структуру завдання
    STask *tsk = (STask *)stas;
    pthread_setspecific(TSYS::sTaskKey, tsk);

    //> Запам'ятовуємо параметри виклику
    void *(*wTask) (void *) = tsk->task;
    void *wTaskArg = tsk->taskArg;

    //> Отримуємо поточні політику і пріоритет
    int policy;
    struct sched_param param;

```

```

pthread_getschedparam(pthread_self(), &policy, &param);
tsk->policy = policy;
tsk->prior = param.sched_priority;

#if __GLIBC_PREREQ(2,4)
//> Отримуємо і завантажуюмо CPU установлення
if(SYS->multCPU() && !(tsk->flgs & STask::Detached))
{
    tsk->cpuSet = TBDS::genDBGet(SYS->nodePath()+"CpuSet:"+tsk->path);
    cpu_set_t cpuset;
    CPU_ZERO(&cpuset);
    string sval;
    bool cpuSetOK = false;
    for(int off = 0; (sval=TSYS::strParse(tsk->cpuSet,0,":",&off)).size();
cpuSetOK = true)
        CPU_SET(atoi(sval.c_str()),&cpuset);
    if(cpuSetOK) pthread_setaffinity_np(pthread_self(), sizeof(cpu_set_t),
&cpuset);
}
    else if(SYS->multCPU() && (tsk->flgs & STask::Detached)) tsk->cpuSet = "NA";
#endif

//> Закінчуємо установки та ініціалізуємо індикатор закінчення
tsk->tid = syscall(SYS_gettid);
tsk->thr = pthread_self();

//> Викликаємо робочі завдання
void *rez = NULL;
try { rez = wTask(wTaskArg); }
catch(TError err)
{
    mess_err(err.cat.c_str(),err.mess.c_str());
    mess_err(SYS->nodePath().c_str(),_("Завдання %u несподівано закінчено
виключенням."),tsk->thr);
}

//> Відмічаємо закінчення завдання
tsk->flgs |= STask::FinishTask;

//> Переміщуємо об'єкти завдання окремо
if(tsk->flgs & STask::Detached) SYS->taskDestroy(tsk->path, NULL);

return rez;
}

void TSYS::taskSleep( int64_t per, time_t cron )
{
    struct timespec sp_tm;
    STask *stsk = (STask*)pthread_getspecific(sTaskKey);

    if(!cron)
    {
        if(!per) per = 1000000000;
        clock_gettime(CLOCK_REALTIME,&sp_tm);
        int64_t end_tm = (int64_t)sp_tm.tv_sec*1000000000+sp_tm.tv_nsec;
        int64_t pnt_tm = (end_tm/per + 1)*per;
        do
        {
            sp_tm.tv_sec = pnt_tm/1000000000; sp_tm.tv_nsec = pnt_tm%1000000000;
            if(clock_nanosleep(CLOCK_REALTIME,TIMER_ABSTIME,&sp_tm,NULL))
            return;
            clock_gettime(CLOCK_REALTIME,&sp_tm);
        }while(((int64_t)sp_tm.tv_sec*1000000000+sp_tm.tv_nsec) < pnt_tm);

        if(stsk)
        {
            stsk->tm_beg = stsk->tm_per;
            stsk->tm_end = end_tm;
            stsk->tm_per = (int64_t)sp_tm.tv_sec*1000000000+sp_tm.tv_nsec;

```

```

    }
}
else
{
    time_t end_tm = time(NULL);
    while(time(NULL) < cron && usleep(1000000) == 0) ;
    if(stsk)
    {
        stsk->tm_beg = stsk->tm_per;
        stsk->tm_end = 100000000011*end_tm;
        stsk->tm_per = 100000000011*time(NULL);
    }
}
}

time_t TSYS::cron( const string &vl, time_t base )
{
    string cronEl, tEl;
    int vbeg, vend, vstep, vm;

    time_t ctm = base?base:time(NULL);
    struct tm ttm;
    localtime_r(&ctm,&ttm);
    ttm.tm_sec = 0;

reload:
    bool isReload = false;

    //> Хвилини check
    cronEl = TSYS::strSepParse(vl,0,' ');
    vm = 200;
    for(int eoff = 0; (tEl=TSYS::strSepParse(cronEl,0,',',&eoff)).size(); )
    {
        vbeg = vend = -1; vstep = 0;
        sscanf(tEl.c_str(), "%d-%d/%d", &vbeg, &vend, &vstep);
        if(vbeg < 0) { sscanf(tEl.c_str(), "%*/%d", &vstep); vbeg=0; vend=59; }
        if(vend < 0) vm = vmin(vm, vbeg+((ttm.tm_min>=vbeg)?60:0));
        else if((vbeg=vmax(0,vbeg)) < (vend=vmin(59,vend)))
        {
            if(ttm.tm_min < vbeg) vm = vmin(vm, vbeg);
            else if((vstep>1 && ttm.tm_min >= (vbeg+((vend-vbeg)/vstep)*vstep)) ||
(vstep <= 0 && ttm.tm_min >= vend))
                vm = vmin(vm, vbeg+60);
            else if(vstep>1 ) vm = vmin(vm, vbeg + vstep*(((ttm.tm_min+1)-
vbeg)/vstep + (((ttm.tm_min+1)-vbeg)%vstep)?1:0));
            else vm = vmin(vm, ttm.tm_min+1);
        }
        if(vm == ttm.tm_min+1) break;
    }
    ttm.tm_min = vm;
    mktime(&ttm);

    //> Перевіряємо час
    cronEl = TSYS::strSepParse(vl,1,' ');
    vm = 200;
    for(int eoff = 0; (tEl=TSYS::strSepParse(cronEl,0,',',&eoff)).size(); )
    {
        vbeg = vend = -1; vstep = 0;
        sscanf(tEl.c_str(), "%d-%d/%d", &vbeg, &vend, &vstep);
        if(vbeg < 0) { sscanf(tEl.c_str(), "%*/%d", &vstep); vbeg=0; vend=23; }
        if(vend < 0) vm = vmin(vm, vbeg+((ttm.tm_hour>vbeg)?24:0));
        else if((vbeg=vmax(0,vbeg)) < (vend=vmin(23,vend)))
        {
            if(ttm.tm_hour < vbeg) vm = vmin(vm, vbeg);
            else if((vstep>1 && ttm.tm_hour > (vbeg+((vend-vbeg)/vstep)*vstep)) ||
(vstep <= 0 && ttm.tm_hour > vend))
                vm = vmin(vm, vbeg+24);
            else if(vstep>1 ) vm = vmin(vm, vbeg + vstep*(((ttm.tm_hour-vbeg)/vstep
+ (((ttm.tm_hour-vbeg)%vstep)?1:0)));

```

```

        else vm = vmin(vm, ttm.tm_hour);
    }
    if(vm == ttm.tm_hour) break;
}
isReload = (vm != 200 && ttm.tm_hour!=vm);
ttm.tm_hour = vm;
mktime(&ttm);
if(isReload) { ttm.tm_min = -1; goto reload; }

//> Перевіряємо день
cronEl = TSYS::strSepParse(vl,2,' ');
string cronElw = TSYS::strSepParse(vl,4,' ');
vm = 200;
if(cronEl != "")
    for(int eoff = 0; (tEl=TSYS::strSepParse(cronEl,0,',',&eof)).size(); )
    {
        vbeg = vend = -1; vstep = 0;
        sscanf(tEl.c_str(),"%d-%d/%d",&vbeg,&vend,&vstep);
        if(vbeg < 0) { sscanf(tEl.c_str(),"*/%d",&vstep); vbeg=1; vend=31; }
        if(vend < 0) vm = vmin(vm,vbeg+((ttm.tm_mday>vbeg)?31:0));
        else if((vbeg=vmax(1,vbeg)) < (vend=vmin(31,vend)))
        {
            if(ttm.tm_mday < vbeg) vm = vmin(vm,vbeg);
            else if((vstep>1 && ttm.tm_mday > (vbeg+((vend-vbeg)/vstep)*vstep)
|| (vstep <= 0 && ttm.tm_mday > vend))
                vm = vmin(vm,vbeg+31);
            else if(vstep>1 ) vm = vmin(vm, vbeg + vstep*((ttm.tm_mday-
vbeg)/vstep + ((ttm.tm_mday-vbeg)%vstep)?1:0));
            else vm = vmin(vm, ttm.tm_mday);
        }
        if(vm == ttm.tm_mday) break;
    }
if(cronEl == "" || (cronElw != "" && !cronElw.empty()))
    for(int eoff = 0; (tEl=TSYS::strSepParse(cronElw,0,',',&eof)).size(); )
    {
        vbeg = vend = -1; vstep = 0;
        sscanf(tEl.c_str(),"%d-%d/%d",&vbeg,&vend,&vstep);
        if(vbeg < 0) { sscanf(tEl.c_str(),"*/%d",&vstep); vbeg=0; vend=6; }
        if(vend < 0) vm = vmin(vm, ttm.tm_mday - ttm.tm_wday +
vbeg+((ttm.tm_wday>vbeg)?7:0));
        else if((vbeg=vmax(0,vbeg)) < (vend=vmin(6,vend)))
        {
            if(ttm.tm_wday < vbeg) vm = vmin(vm, ttm.tm_mday - ttm.tm_wday +
vbeg);
            else if((vstep>1 && ttm.tm_wday > (vbeg+((vend-vbeg)/vstep)*vstep)
|| (vstep <= 0 && ttm.tm_wday > vend))
                vm = vmin(vm, ttm.tm_mday - ttm.tm_wday + vbeg+7);
            else if(vstep>1) vm = vmin(vm, ttm.tm_mday - ttm.tm_wday + vbeg +
vstep*((ttm.tm_wday-vbeg)/vstep + ((ttm.tm_wday-vbeg)%vstep)?1:0));
            else vm = vmin(vm, ttm.tm_mday);
        }
        if(vm == ttm.tm_mday) break;
    }
isReload = (vm!=200 && ttm.tm_mday!=vm);
if(vm <= 31) ttm.tm_mday = vm;
else { ttm.tm_mday = vm-31; ttm.tm_mon++; }
mktime(&ttm);
if(isReload) { ttm.tm_min = -1; ttm.tm_hour = 0; goto reload; }

//> Перевіряємо місяць
cronEl = TSYS::strSepParse(vl,3,' ');
vm = 200;
for(int eoff = 0; (tEl=TSYS::strSepParse(cronEl,0,',',&eof)).size(); )
{
    vbeg = vend = -1; vstep = 0;
    sscanf(tEl.c_str(),"%d-%d/%d",&vbeg,&vend,&vstep);
    if(vbeg < 0) { sscanf(tEl.c_str(),"*/%d",&vstep); vbeg=1; vend=12; }
    if(vend < 0) vm = vmin(vm,vbeg+((ttm.tm_mon+1)>vbeg)?12:0);
    else if((vbeg=vmax(1,vbeg)) < (vend=vmin(12,vend)))

```

```

    {
        if((ttm.tm_mon+1) < vbeg) vm = vmin(vm,vbeg);
        else if((vstep>1 && (ttm.tm_mon+1) > (vbeg+((vend-vbeg)/vstep)*vstep))
|| (vstep <= 0 && (ttm.tm_mon+1) > vend))
            vm = vmin(vm,vbeg+12);
        else if(vstep>1) vm = vmin( vm, vbeg + vstep*(((ttm.tm_mon+1)-
vbeg)/vstep + (((ttm.tm_mon+1)-vbeg)%vstep)?1:0));
        else vm = vmin(vm, ttm.tm_mon+1);
    }
    if(vm == (ttm.tm_mon+1)) break;
}
isReload = (vm!=200 && ttm.tm_mon!=(vm-1));
ttm.tm_mon = vm-1;
mktime(&ttm);
if(isReload) { ttm.tm_min = -1; ttm.tm_hour = 0; ttm.tm_mday = 1; goto
reload; }

return mktime(&ttm);
}

```

```

TVariant TSYS::objFuncCall( const string &iid, vector<TVariant> &prms, const
string &user )
{
    // int message(string cat, int level, string mess) - форматуємо системне
повідомлення <mess> з категорією <cat>, рівнем <level>
    // cat - повідомлення категорії
    // level - повідомлення рівня
    // mess - повідомлення тексту
    if(iid == "message" && prms.size() >= 3) { message(
prms[0].getS().c_str(), (TMess::Type)prms[1].getI(), "%s",
prms[2].getS().c_str() ); return 0; }
    // int messDebug(string cat, string mess) - форматуємо системне повідомлення
<mess> з категорією <cat> і відповідний рівень
    // cat - повідомлення категорії
    // mess - повідомлення тексту
    if(iid == "messDebug" && prms.size() >= 2) { mess_debug(
prms[0].getS().c_str(), "%s", prms[1].getS().c_str() ); return 0; }
    if(iid == "messInfo" && prms.size() >= 2) { mess_info(
prms[0].getS().c_str(), "%s", prms[1].getS().c_str() ); return 0; }
    if(iid == "messNote" && prms.size() >= 2) { mess_note(
prms[0].getS().c_str(), "%s", prms[1].getS().c_str() ); return 0; }
    if(iid == "messWarning" && prms.size() >= 2){ mess_warning(
prms[0].getS().c_str(), "%s", prms[1].getS().c_str() ); return 0; }
    if(iid == "messErr" && prms.size() >= 2) { mess_err(
prms[0].getS().c_str(), "%s", prms[1].getS().c_str() ); return 0; }
    if(iid == "messCrit" && prms.size() >= 2) { mess_crit(
prms[0].getS().c_str(), "%s", prms[1].getS().c_str() ); return 0; }
    if(iid == "messAlert" && prms.size() >= 2) { mess_alert(
prms[0].getS().c_str(), "%s", prms[1].getS().c_str() ); return 0; }
    if(iid == "messEmerg" && prms.size() >= 2) { mess_emerg(
prms[0].getS().c_str(), "%s", prms[1].getS().c_str() ); return 0; }
    // string system(string cmd, bool noPipe = false) - викликаємо консольні
команди <cmd> для повернення у ОС результатів з каналів
    // cmd - текс команди
    // noPipe - результат блокують для другорядного виклику
    if(iid == "system" && prms.size() >= 1)
    {
        if(prms.size() >= 2 && prms[1].getB()) return
system(prms[0].getS().c_str());
        FILE *fp = popen(prms[0].getS().c_str(),"r");
        if(!fp) return string("");

        char buf[STR_BUF_LEN];
        string rez;
        for(int r_cnt = 0; (r_cnt=fread(buf,1,sizeof(buf),fp)); )
            rez.append(buf,r_cnt);

        pclose(fp);
        return rez;
    }
}

```

```

}
// string fileRead( string file ) - Повертаємо <file> контент у рядку.
if(iid == "fileRead" && prms.size() >= 1)
{
    char buf[STR_BUF_LEN];
    string rez;
    int hd = open(prms[0].getS().c_str(),O_RDONLY);
    if(hd != -1)
    {
        for(int len = 0; (len=read(hd,buf,sizeof(buf))) > 0; )
rez.append(buf,len);
        close(hd);
    }
    return rez;
}
// int fileWrite( string file, string str, bool append = false ) - Записуємо
<str> до <file>, переміщуємо представлення, або <append>.
// Return wrote bytes count.
if(iid == "fileWrite" && prms.size() >= 2)
{
    int wcnt = 0, wflags = O_WRONLY|O_CREAT|O_TRUNC;
    string val = prms[1].getS();
    if(prms.size() >= 3 && prms[2].getB()) wflags = O_WRONLY|O_CREAT|O_APPEND;
    int hd = open(prms[0].getS().c_str(), wflags, 0664);
    if(hd != -1)
    {
        wcnt = write(hd,val.data(),val.size());
        close(hd);
    }
    return wcnt;
}
// XMLNodeObj XMLNode(string name = "") - створюємо XML об'єкт вузлів з
іменем <name>
// name - XML ім'я вузлу
if(iid == "XMLNode") return new XMLNodeObj((prms.size()>=1) ? prms[0].getS()
: "");
// string cntrReq(XMLNodeObj req, string stat = "") - запит інтерфейсу
управління до системи через XML
// req - запити XML вузлів
// stat - переміщуємо WebCisco_Application_Centric_Infrastructure-робочу
станцію для запиту
if(iid == "cntrReq" && prms.size() >= 1)
{
    XMLNode req;
    if(!dynamic_cast<XMLNodeObj*>(prms[0].getO())) return string(_("1:Запит не
об'єктний!"));
    ((XMLNodeObj*)prms[0].getO())->toXMLNode(req);
    string path = req.attr("path");
    if(prms.size() < 2 || prms[1].getS().empty())
    {
        req.setAttr("user",user);
        cntrCmd(&req);
    }
    else
    {
        req.setAttr("path","/"+prms[1].getS()+path);
        transport().at().cntrIfCmd(req,"cntrReq");
        req.setAttr("path",path);
    }
    ((XMLNodeObj*)prms[0].getO())->fromXMLNode(req);
    return string("0");
}
// string sleep(int tm, int ntm = 0) - викликаємо завдання переходу до
сплячого режиму через <tm> секунд та <ntm> наносекунд.
// tm - чекаємо цей час у секундах
// ntm - чекаємо цей час у наносекундах
if(iid == "sleep" && prms.size() >= 1)
{
    struct timespec sp_tm;

```

```

    sp_tm.tv_sec = prms[0].getI();
    sp_tm.tv_nsec = (prms.size() >= 2) ? prms[1].getI() : 0;
    int rez = clock_nanosleep(CLOCK_REALTIME,0,&sp_tm,NULL);
    return rez;
}
// int time(int usec) - повертаємо абсолютний час у секундах від 1/1/1970 та
в мікросекундах, якщо <usec> задано
// usec - мікросекунди of time
if(iid == "time")
{
    if(prms.empty()) return (int)time(NULL);
    int64_t tm = curTime();
    prms[0].setI(tm%1000000); prms[0].setModify();
    return (int)(tm/1000000);
}
// int localtime(int fullsec, int sec, int min, int hour, int mday, int
month, int year, int wday, int yday, int isdst)
// - повертаємо повну дату, базуємо на абсолютному часі у секундах
<fullsec> від 1.1.1970
// fullsec - час джерела у секундах від 1.1.1970
// sec - секунди
// min - хвилини
// hour - часи
// mday - дні місяця
// month - місяці
// year - роки
// wday - дні неділі
// yday - дні року
// isdst - відмітка про літній час
if(iid == "localtime" && prms.size() >= 2)
{
    time_t tm_t = prms[0].getI();
    struct tm tm_tm;
    localtime_r(&tm_t,&tm_tm);

    prms[1].setI(tm_tm.tm_sec); prms[1].setModify();
    if(prms.size() >= 3) { prms[2].setI(tm_tm.tm_min); prms[2].setModify();
}
    if(prms.size() >= 4) { prms[3].setI(tm_tm.tm_hour);
prms[3].setModify(); }
    if(prms.size() >= 5) { prms[4].setI(tm_tm.tm_mday);
prms[4].setModify(); }
    if(prms.size() >= 6) { prms[5].setI(tm_tm.tm_mon); prms[5].setModify();
}
    if(prms.size() >= 7) { prms[6].setI(1900+tm_tm.tm_year);
prms[6].setModify(); }
    if(prms.size() >= 8) { prms[7].setI(tm_tm.tm_wday);
prms[7].setModify(); }
    if(prms.size() >= 9) { prms[8].setI(tm_tm.tm_yday);
prms[8].setModify(); }
    if(prms.size() >= 10) { prms[9].setI(tm_tm.tm_isdst);
prms[9].setModify(); }
    return 0;
}
// string strftime(int sec, string form = "%Y-%m-%d %H:%M:%S") - перетворює
абсолютний час <sec> у рядок формату <form>
// sec - час у секундах від 1.1.1970
// form - вихідний форматований рядок
if(iid == "strftime" && !prms.empty())
{
    time_t tm_t = prms[0].getI();
    struct tm tm_tm;
    localtime_r(&tm_t,&tm_tm);
    char buf[1000];
    int rez = strftime(buf, sizeof(buf), (prms.size()>=2) ?
prms[1].getS().c_str() : "%Y-%m-%d %H:%M:%S", &tm_tm);
    return (rez>0) ? string(buf,rez) : "";
}

```

```

// int strtptime(string str, string form = "%Y-%m-%d %H:%M:%S") - повертає
час у секундах від 01/01/1970,
// базується на запису рядку часу <str>, відповідно до вказаного
шаблону <form>
// str - джерело часу у рядку
// form - рядки часу у форматі POSIX-функцій "strptime"
if(iid == "strptime" && !prms.empty())
{
    struct tm stm;
    stm.tm_isdst = -1;
    strtptime(prms[0].getS().c_str(), (prms.size()>=2) ? prms[1].getS().c_str()
: "%Y-%m-%d %H:%M:%S", &stm);
    return (int)mkttime(&stm);
}
// int cron(string cronreq, int base = 0) - повертає час , планується у
форматі стандартного Cron <cronreq>,
// початок від основного часу<base> або від поточного, якщо основа не
вказана
// cronreq - розповсюджений у стандартному форматі Cron
// base - основний час
if(iid == "cron" && !prms.empty())
    return (int)cron(prms[0].getS(), (prms.size()>=2) ? prms[1].getI() : 0);
// string strFromCharCode(int char1, int char2, int char3, ...) - створює
рядок з кодових символів
// char1, char2, char3 - кодові символи
if(iid == "strFromCharCode")
{
    string rez;
    for(unsigned i_p = 0; i_p < prms.size(); i_p++)
        rez += (unsigned char)prms[i_p].getI();
    return rez;
}
// string strCodeConv( string src, string fromCP, string toCP ) - Текстовий
рядок перекодує з кодової сторінки <fromCP> до кодової сторінки <toCP>.
// src - source text;
// fromCP - з кодової сторінки , порожньої для внутрішньої кодової сторінки
;
// toCP - до кодової сторінки , порожньої для внутрішньої кодової сторінки
.
if(iid == "strCodeConv" && prms.size() >= 3)
    return Mess->codeConv((prms[1].getS().size() ? prms[1].getS() : Mess-
>charset()),
        (prms[2].getS().size() ? prms[2].getS() : Mess->charset()),
prms[0].getS());

return TCntrNode::objFuncCall(iid,prms,user);
}

void TSYS::cntrCmdProc( XMLNode *opt )
{
    char buf[STR_BUF_LEN];

    //Беремо сторінку інформації
    if(opt->name() == "info")
    {
        TCntrNode::cntrCmdProc(opt);
        snprintf(buf, sizeof(buf), _("%s station:
\\%s\\"), PACKAGE_NAME, name().c_str());
        ctrMkNode("Cisco_Application_Centric_Infrastructure_cntr",opt,-
1, "/", buf, R_R_R_);
        if(ctrMkNode("branches",opt,-1, "/br", "", R_R_R_))
            ctrMkNode("grp",opt,-
1, "/br/sub_", _("Subsystem"), R_R_R_, "root", "root", 1, "idm", "1");
        if(TUIS::icoPresent(id())) ctrMkNode("img",opt,-1, "/ico", "", R_R_R_);
        if(ctrMkNode("area",opt,-1, "/gen", _("Station"), R_R_R_))
        {
            ctrMkNode("fld",opt,-
1, "/gen/id", _("ID"), R_R_R_, "root", "root", 1, "tp", "str");

```

```

ctrMkNode("fld",opt,-
1,"/gen/stat",_("Station"),RWRWR_,"root","root",1,"tp","str");
ctrMkNode("fld",opt,-
1,"/gen/prog",_("Program"),R_R_R_,"root","root",1,"tp","str");
ctrMkNode("fld",opt,-
1,"/gen/ver",_("Version"),R_R_R_,"root","root",1,"tp","str");
ctrMkNode("fld",opt,-1,"/gen/host",_("Ім'я
хосту"),R_R_R_,"root","root",1,"tp","str");
ctrMkNode("fld",opt,-1,"/gen/user",_("Користувач
системи"),R_R_R_,"root","root",1,"tp","str");
ctrMkNode("fld",opt,-1,"/gen/sys",_("Операційна
система"),R_R_R_,"root","root",1,"tp","str");
ctrMkNode("fld",opt,-1,"/gen/frq",_("Частота
(MHZ)"),R_R_R_,"root","root",1,"tp","real");
ctrMkNode("fld",opt,-1,"/gen/clk_res",_("Значення годинника реального
часу"),R_R_R_,"root","root",1,"tp","str");
ctrMkNode("fld",opt,-1,"/gen/in_charset",_("Внутрішній набір
символів"),R_R_,"root","root",1,"tp","str");
ctrMkNode("fld",opt,-1,"/gen/config",_("Конфігураційний
файл"),R_R_,"root","root",1,"tp","str");
ctrMkNode("fld",opt,-1,"/gen/workdir",_("Робоча
директорія"),RWRW_,"root","root",1,"tp","str");
ctrMkNode("fld",opt,-1,"/gen/icodir",_("Директорія
іконок"),RWRW_,"root","root",1,"tp","str");
ctrMkNode("fld",opt,-1,"/gen/moddir",_("Директорія
модулів"),RWRW_,"root","root",1,"tp","str");
ctrMkNode("fld",opt,-1,"/gen/wrk_db",_("Робоча база
даних"),RWRWR_,"root","root",4,"tp","str","dest","select","select","/db/list",
"help",_("Адрес робочої бази даних у форматі [<БД module>.<БД
name>].\n Змінюємо ці поля якщо необхідно зберегти або завантажити усю систему з
іншої БД.));
ctrMkNode("fld",opt,-1,"/gen/saveExit",_("Збереження змін у системі
перед виходом"),RWRWR_,"root","root",2,"tp","bool",
"help",_("Обираємо автоматичне збереження системи до БД перед
виходом.));
ctrMkNode("fld",opt,-1,"/gen/savePeriod",_("Збереження періоду роботи
у системі "),RWRWR_,"root","root",2,"tp","dec",
"help",_("Використовуємо нульовий період (секунди) для періодичного
збереження змін частин системи у БД.));
ctrMkNode("fld",opt,-
1,"/gen/lang",_("Language"),RWRWR_,"root","root",1,"tp","str");
ctrMkNode("fld",opt,-1,"/gen/baseLang",_("Базова мова тексту
змінних"),RWRWR_,"root","root",5,"tp","str","len","2","dest","sel_ed","select",
"/gen/baseLangLs",
"help",_("Мультимовність для змінного тексту підтримки для вибору
базової мови.));
if(ctrMkNode("area",opt,-1,"/gen/mess",_("Повідомлення"),R_R_R_))
{
ctrMkNode("fld",opt,-1,"/gen/mess/lev",_("Найменший
рівень"),RWRWR_,"root","root",3,
"tp","dec","len","1","help",_("Повідомлення найменшого рівня для
відображення процесів, які відбуваються у системі.));
ctrMkNode("fld",opt,-1,"/gen/mess/log_sysl",_("To
syslog"),RWRWR_,"root","root",1,"tp","bool");
ctrMkNode("fld",opt,-1,"/gen/mess/log_stdo",_("To
stdout"),RWRWR_,"root","root",1,"tp","bool");
ctrMkNode("fld",opt,-1,"/gen/mess/log_stde",_("To
stderr"),RWRWR_,"root","root",1,"tp","bool");
ctrMkNode("fld",opt,-1,"/gen/mess/log_arch",_("До
архіву"),RWRWR_,"root","root",1,"tp","bool");
}
}
if(ctrMkNode("area",opt,-1,"/subs",_("Підсистема")))
ctrMkNode("list",opt,-
1,"/subs/br",_("Підсистеми"),R_R_R_,"root","root",3,"idm","1","tp","br","br_pref",
"sub_");
if(ctrMkNode("area",opt,-1,"/tasks",_("Tasks"),R_R_))
if(ctrMkNode("table",opt,-
1,"/tasks/tasks",_("Завдання"),RWRW_,"root","root",2,"key","path",

```

```

    "help",!multCPU()?"":_("Для CPU встановлюємо рядок номерів
процесорів використання, відокремлений символом:'.\n"
"CPU починаємо з 0.)))
{
    ctrMkNode("list",opt,-
1, "/tasks/tasks/path",_("Path"),R_R____,"root","root",1,"tp","str");
    ctrMkNode("list",opt,-
1, "/tasks/tasks/thrd",_("Поток"),R_R____,"root","root",1,"tp","str");
    ctrMkNode("list",opt,-
1, "/tasks/tasks/tid",_("TID"),R_R____,"root","root",1,"tp","dec");
    ctrMkNode("list",opt,-
1, "/tasks/tasks/stat",_("Статус"),R_R____,"root","root",1,"tp","str");
    ctrMkNode("list",opt,-
1, "/tasks/tasks/plc",_("Політика"),R_R____,"root","root",1,"tp","str");
    ctrMkNode("list",opt,-
1, "/tasks/tasks/prior",_("Prior."),R_R____,"root","root",1,"tp","dec");
#ifdef __GLIBC_PREREQ(2,4)
    if(multCPU())
        ctrMkNode("list",opt,-1, "/tasks/tasks/cpuSet",_("CPU
встановлення"),RWRW____,"root","root",1,"tp","str");
#endif
}
if( !cntrEmpty() && ctrMkNode("area",opt,-1, "/cntr",_("Країна")) )
    if( ctrMkNode("table",opt,-
1, "/cntr/cntr",_("Counters"),R_R____,"root","root") )
    {
        ctrMkNode("list",opt,-
1, "/cntr/cntr/id","ID",R_R____,"root","root",1,"tp","str");
        ctrMkNode("list",opt,-
1, "/cntr/cntr/vl",_("Value"),R_R____,"root","root",1,"tp","real");
    }
    if( ctrMkNode("area",opt,-1, "/hlp",_("Help"),R_R____) )
        ctrMkNode("fld",opt,-1, "/hlp/g_help",_("Опції
допомоги"),R_R____,"root","root",3,"tp","str","cols","90","rows","10");
    return;
}

//Процес управління на сторінці
string a_path = opt->attr("path");
if(a_path == "/ico" && ctrChkNode(opt))
{
    string itp;
    opt->setText(TSYS::strEncode(TUIS::icoGet(id()),&itp),TSYS::base64);
    opt->setAttr("tp",itp);
}
else if(a_path == "/gen/host" && ctrChkNode(opt)) opt->setText(host());
else if(a_path == "/gen/sys" && ctrChkNode(opt))
{
    utsname ubuf; uname(&ubuf);
    opt->setText(string(ubuf.sysname)+"-"+ubuf.release);
}
else if(a_path == "/gen/user" && ctrChkNode(opt)) opt->setText(mUser);
else if(a_path == "/gen/prog" && ctrChkNode(opt)) opt->setText(PACKAGE_NAME);
else if(a_path == "/gen/ver" && ctrChkNode(opt)) opt->setText(VERSION);
else if(a_path == "/gen/id" && ctrChkNode(opt)) opt->setText(id());
else if(a_path == "/gen/stat")
{
    if(ctrChkNode(opt,"get",RWRWR____,"root","root",SEC_RD)) opt->setText(name());
    if(ctrChkNode(opt,"set",RWRWR____,"root","root",SEC_WR)) setName(opt->text());
}
else if(a_path == "/gen/frq" && ctrChkNode(opt)) opt->
setText(TSYS::real2str((float)sysClk()/1000000.,6));
else if(a_path == "/gen/clk_res" && ctrChkNode(opt))
{
    struct timespec tmval;
    clock_getres(CLOCK_REALTIME,&tmval);
    opt->setText(TSYS::time2str(1e-3*tmval.tv_nsec));//
TSYS::real2str((float)tmval.tv_nsec/1000000.,4));
}
}

```

```

    else if(a_path == "/gen/in_charset" && ctrChkNode(opt)) opt->setText(Mess-
>charset());
    else if(a_path == "/gen/config" && ctrChkNode(opt))          opt-
>setText(mConfFile);
    else if(a_path == "/gen/wrk_db" )
    {
        if(ctrChkNode(opt,"get",RWRWR_,"root","root",SEC_RD)) opt-
>setText(mWorkDB);
        if(ctrChkNode(opt,"set",RWRWR_,"root","root",SEC_WR)) setWorkDB(opt-
>text());
    }
    else if(a_path == "/gen/saveExit")
    {
        if(ctrChkNode(opt,"get",RWRWR_,"root","root",SEC_RD)) opt->setText(
int2str(saveAtExit() ) );
        if(ctrChkNode(opt,"set",RWRWR_,"root","root",SEC_WR)) setSaveAtExit(
atoi(opt->text().c_str() ) );
    }
    else if(a_path == "/gen/savePeriod")
    {
        if(ctrChkNode(opt,"get",RWRWR_,"root","root",SEC_RD)) opt->setText(
int2str(savePeriod() ) );
        if(ctrChkNode(opt,"set",RWRWR_,"root","root",SEC_WR)) setSavePeriod(
atoi(opt->text().c_str() ) );
    }
    else if(a_path == "/gen/workdir")
    {
        if(ctrChkNode(opt,"get",R_R___,"root","root",SEC_RD)) opt-
>setText(workDir());
        if(ctrChkNode(opt,"set",R_R___,"root","root",SEC_WR)) setWorkDir(opt-
>text().c_str());
    }
    else if(a_path == "/gen/icodir")
    {
        if(ctrChkNode(opt,"get",R_R___,"root","root",SEC_RD)) opt-
>setText(icoDir());
        if(ctrChkNode(opt,"set",R_R___,"root","root",SEC_WR)) setIcoDir(opt-
>text().c_str());
    }
    else if(a_path == "/gen/moddir")
    {
        if(ctrChkNode(opt,"get",R_R___,"root","root",SEC_RD)) opt-
>setText(modDir());
        if(ctrChkNode(opt,"set",R_R___,"root","root",SEC_WR)) setModDir(opt-
>text().c_str());
    }
    else if(a_path == "/gen/lang")
    {
        if(ctrChkNode(opt,"get",RWRWR_,"root","root",SEC_RD)) opt->setText(Mess-
>lang());
        if(ctrChkNode(opt,"set",RWRWR_,"root","root",SEC_WR)) Mess->setLang(opt-
>text());
    }
    else if(a_path == "/gen/baseLang")
    {
        if(ctrChkNode(opt,"get",RWRWR_,"root","root",SEC_RD)) opt->setText(Mess-
>lang2CodeBase());
        if(ctrChkNode(opt,"set",RWRWR_,"root","root",SEC_WR)) Mess-
>setLang2CodeBase(opt->text());
    }
    else if(a_path == "/gen/baseLangLs" && ctrChkNode(opt))
    {
        opt->childAdd("el")->setText(Mess->lang2Code());
        if(!Mess->lang2CodeBase().empty() && Mess->lang2CodeBase() != Mess-
>lang2Code())
            opt->childAdd("el")->setText(Mess->lang2CodeBase());
        opt->childAdd("el")->setText("");
    }
    else if(a_path == "/gen/mess/lev")

```

```

    {
        if(ctrChkNode(opt,"get",RWRWR_,"root","root",SEC_RD)) opt->
        >setText(TSYS::int2str(Mess->messLevel()));
        if(ctrChkNode(opt,"set",RWRWR_,"root","root",SEC_WR)) Mess->
        >setMessLevel(atoi(opt->text().c_str()));
    }
    else if(a_path == "/gen/mess/log_sysl")
    {
        if(ctrChkNode(opt,"get",RWRWR_,"root","root",SEC_RD)) opt->setText((Mess->
        >logDirect() &0x01)?"1":"0");
        if(ctrChkNode(opt,"set",RWRWR_,"root","root",SEC_WR)) Mess->setLogDirect(
        atoi(opt->text().c_str())?Mess->logDirect()|0x01:Mess->logDirect() &(~0x01) );
    }
    else if(a_path == "/gen/mess/log_stdio")
    {
        if(ctrChkNode(opt,"get",RWRWR_,"root","root",SEC_RD)) opt->setText((Mess->
        >logDirect() &0x02)?"1":"0");
        if(ctrChkNode(opt,"set",RWRWR_,"root","root",SEC_WR)) Mess->setLogDirect(
        atoi(opt->text().c_str())?Mess->logDirect()|0x02:Mess->logDirect() &(~0x02) );
    }
    else if(a_path == "/gen/mess/log_stde")
    {
        if(ctrChkNode(opt,"get",RWRWR_,"root","root",SEC_RD)) opt->setText((Mess->
        >logDirect() &0x04)?"1":"0");
        if(ctrChkNode(opt,"set",RWRWR_,"root","root",SEC_WR)) Mess->setLogDirect(
        atoi(opt->text().c_str())?Mess->logDirect()|0x04:Mess->logDirect() &(~0x04) );
    }
    else if(a_path == "/gen/mess/log_arch")
    {
        if(ctrChkNode(opt,"get",RWRWR_,"root","root",SEC_RD)) opt->setText((Mess->
        >logDirect() &0x08)?"1":"0");
        if(ctrChkNode(opt,"set",RWRWR_,"root","root",SEC_WR)) Mess->setLogDirect(
        atoi(opt->text().c_str())?Mess->logDirect()|0x08:Mess->logDirect() &(~0x08) );
    }
    else if((a_path == "/br/sub_" || a_path == "/subs/br") &&
    ctrChkNode(opt,"get",R_R_R_,"root","root",SEC_RD))
    {
        vector<string> lst;
        list(lst);
        for(unsigned i_a=0; i_a < lst.size(); i_a++)
            opt->childAdd("el")->setAttr("id",lst[i_a])->
            >setText(at(lst[i_a]).at().subName());
    }
    else if(a_path == "/tasks/tasks")
    {
        if(ctrChkNode(opt,"get",RWRW_,"root","root"))
        {
            XMLNode *n_path = ctrMkNode("list",opt,-
            1,"/tasks/tasks/path","",R_R_,"root","root");
            XMLNode *n_thr = ctrMkNode("list",opt,-
            1,"/tasks/tasks/thrd","",R_R_,"root","root");
            XMLNode *n_tid = ctrMkNode("list",opt,-
            1,"/tasks/tasks/tid","",R_R_,"root","root");
            XMLNode *n_stat = ctrMkNode("list",opt,-
            1,"/tasks/tasks/stat","",R_R_,"root","root");
            XMLNode *n_plc = ctrMkNode("list",opt,-
            1,"/tasks/tasks/plc","",R_R_,"root","root");
            XMLNode *n_prior = ctrMkNode("list",opt,-
            1,"/tasks/tasks/prior","",R_R_,"root","root");
            XMLNode *n_cpuSet = (multCPU() ? ctrMkNode("list",opt,-
            1,"/tasks/tasks/cpuSet","",RWRW_,"root","root") : NULL);

            ResAlloc res(taskRes,false);
            for(map<string,STask>::iterator it = mTasks.begin(); it !=
            mTasks.end(); it++)
            {
                if(n_path) n_path->childAdd("el")->setText(it->first);
                if(n_thr) n_thr->childAdd("el")->setText(TSYS::uint2str(it->
                >second.thr));
            }
        }
    }
}

```

```

        if(n_tid) n_tid->childAdd("el")->setText(TSYS::int2str(it-
>second.tid));
        if(n_stat)
        {
            int64_t tm_beg = 0, tm_end = 0, tm_per = 0;
            for(int i_tr = 0; tm_beg == tm_per && i_tr < 2; i_tr++)
            { tm_beg = it->second.tm_beg; tm_end = it->second.tm_end; tm_per
= it->second.tm_per; }
            XMLNode *cn = n_stat->childAdd("el");
            if(it->second.flgs&STask::FinishTask) cn->setText(_("Закінчено.
"));
            if(tm_beg && tm_beg < tm_per)
                cn->setText(cn->text()+TSYS::strMess(_("Останій: %s.
Завантажено: %3.1f% (%s з %s)"),
                    time2str((time_t)(1e-9*tm_per),"%d-%m-%Y
%H:%M:%S").c_str(), 100*(double)(tm_end-tm_beg)/(double)(tm_per-tm_beg),
                    time2str(1e-3*(tm_end-tm_beg)).c_str(), time2str(1e-
3*(tm_per-tm_beg)).c_str()));
        }
        if(n_plc)
        {
            string plcVl = _("Стандартний");
            if(it->second.policy == SCHED_RR) plcVl = _("Кругова система
");
            #if __GLIBC_PREREQ(2,4)
                if(it->second.policy == SCHED_BATCH) plcVl = _("Style
\"batch\"");
            #endif
            n_plc->childAdd("el")->setText(plcVl);
        }
        if(n_prior) n_prior->childAdd("el")->setText(TSYS::int2str(it-
>second.prior));
        if(n_cpuSet) n_cpuSet->childAdd("el")->setText(it-
>second.cpuSet);
    }
}
#if __GLIBC_PREREQ(2,4)
    if(multCPU() && ctrChkNode(opt,"set",RWRW_,"root","root",SEC_WR) && opt-
>attr("col") == "cpuSet")
    {
        ResAlloc res(taskRes,true);
        map<string,STask>::iterator it = mTasks.find(opt->attr("key_path"));
        if(it == mTasks.end()) throw TError(nodePath().c_str(),_("Не
представлене завдання '%s'."));
        if(it->second.flgs & STask::Detached) return;

        it->second.cpuSet = opt->text();

        cpu_set_t cpuset;
        CPU_ZERO(&cpuset);
        string sval;
        for(int off = 0; (sval=TSYS::strParse(it-
>second.cpuSet,0,":",&off)).size(); )
            CPU_SET(atoi(sval.c_str()),&cpuset);
        int rez = pthread_setaffinity_np(it->second.thr, sizeof(cpu_set_t),
&cpuset);
        res.release();
        TBDS::genDBSet(nodePath()+"CpuSet:"+it->first,opt->text());
        if(rez == EINVAL && opt->text().size()) throw
TError(nodePath().c_str(),_("Не встановлено жодних дозволених процесорів."));
        if(rez && opt->text().size()) throw TError(nodePath().c_str(),_("CPU
установки для потоку помилкові."));
    }
#endif
}
    if(!cntrEmpty() && a_path == "/cntr/cntr" &&
ctrChkNode(opt,"get",R_R_,"root","root"))
    {

```

```

XMLNode *n_id      = ctrMkNode("list",opt,-
1,"/cntr/cntr/id","",R_R___,"root","root");
XMLNode *n_vl      = ctrMkNode("list",opt,-
1,"/cntr/cntr/vl","",R_R___,"root","root");

ResAlloc res( nodeRes(), false );
for(map<string,double>::iterator icnt = mCntrs.begin(); icnt !=
mCntrs.end(); icnt++)
{
    if(n_id)      n_id->childAdd("el")->setText(icnt->first);
    if(n_vl)      n_vl->childAdd("el")->setText(TSYS::real2str(icnt-
>second));
}
}
else if(a_path == "/hlp/g_help" &&
ctrChkNode(opt,"get",R_R___,"root","root",SEC_RD)) opt->setText(optDescr());
else TCntrNode::cntrCmdProc(opt);
}

```

K6П3_2024