

Центральноукраїнський національний технічний університет
Центр заочної та дистанційної освіти
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“ ____ ” _____ 2023 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи моніторингу
центрів обробки даних, побудованих з використанням мереж
NGBaseT”

Виконав здобувач вищої освіти
II курсу, групи КН-22МЗ
ОПП «Комп’ютерні науки»
спеціальності 122 «Комп’ютерні науки»
_____ Маляренко Р.В.
« ____ » _____ 2023 р.

Керівник проекту
кандидат технічних наук
_____ Лисенко І.А.
« ____ » _____ 2023 р.
Рецензент _____

Центральноукраїнський національний технічний університет
Центр *Заочної та дистанційної освіти*
Кафедра *Кібербезпеки та програмного забезпечення*
Рівень вищої освіти *магістр*
Галузь знань 12 *“Інформаційні технології”*
Спеціальність 122 *“Комп’ютерні науки”*
Освітньо-професійна (освітньо-наукова) програма *“Комп’ютерні науки”*

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 6 » вересня 2023 року

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

Маляренку Роману Віталійовичу

(прізвище, ім'я, по батькові)

- Тема роботи *Дослідження та програмна реалізація системи моніторингу центрів обробки даних, побудованих з використанням мереж NGBaseT*
- Керівник роботи *Лисенко Ірина Анатоліївна, канд. техн. наук*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу № 37-13 від 04.08.2023 року
- Строк подання студентом роботи до захисту 10.12.2023 р.
- Мета та завдання випускної кваліфікаційної роботи: *Метою розробки є дослідження та програмна реалізація системи моніторингу центрів обробки даних, побудованих з використанням мереж NGBaseT*
- Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
 - Призначення та область використання.*
 - Наукова новизна.*
 - Перегляд аналогічних існуючих систем.*
 - Економічна ефективність розробленої програми.*
 - Опис і обґрунтування проектних рішень.*
 - Заходи з охорони праці та техніки безпеки.*
 - Етапи програмування системи.*
 - Висновки.*
 - Впровадження системи в промислову експлуатацію*
- Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

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

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Савеленко Г.В.	05.10.2023	14.11.2023
Охорона праці	Оришака О.В.	06.10.2023	16.11.2023

7. Дата видачі завдання « 6 » вересня 2023 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	10.10.2023 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2023 р.	
3.	Розробка моделі компонента	20.10.2023 р.	
4.	Розробка структур даних	25.10.2023 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2023 р.	
6.	Програмування алгоритмів	10.11.2023 р.	
7.	Розрахунок економічної ефективності	13.11.2023 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2023 р.	
9.	Оформлення ПЗ	17.11.2023 р.	
10.	Попередній захист роботи	10.12.2023 р.	

Дата видачі завдання
« 6 » вересня 2023 р.

Підпис керівника

_____ (прізвище та ініціали)

Завдання прийнято до виконання
« 6 » вересня 2023 р.

Підпис здобувача

_____ (прізвище та ініціали)

АНОТАЦІЯ

Маляренко Р.В. Дослідження та програмна реалізація системи моніторингу центрів обробки даних, побудованих з використанням мереж NGBaseT. 122 Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2023.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи моніторингу центрів обробки даних, побудованих з використанням мереж NGBaseT.

Метою розробки є дослідження та програмна реалізація системи моніторингу центрів обробки даних, побудованих з використанням мереж NGBaseT.

Об'єктом дослідження є процес моніторингу центрів обробки даних, побудованих з використанням мереж NGBaseT.

Предметом дослідження є методи моніторингу центрів обробки даних, побудованих з використанням мереж NGBaseT.

Методи дослідження базуються на методах хмарних технологій, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи моніторингу центрів обробки даних, побудованих з використанням мереж NGBaseT.

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 архітектури IBM PC з ОС Windows 10/11.

Програму розроблено в середовищі Delphi 10.4.1.

Ключові слова: комп'ютерні науки, NGBaseT

ABSTRACT

Maliarenko R.V. Research and software implementation of a monitoring system for data centers built using NGBaseT networks. 122 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2023.

In this graduation thesis for the second (master's) level of higher education, software is developed, which is intended for the monitoring system of data processing centers built using NGBaseT networks.

The purpose of the development is the research and software implementation of the monitoring system of data centers built using NGBaseT networks.

The object of the study is the process of monitoring data centers built using NGBaseT networks.

The subject of the research is methods of monitoring data centers built using NGBaseT networks.

Research methods are based on methods of cloud technologies, methods of mathematical statistics, methods of software development.

The result of the work is the software implementation of the data center monitoring system built using NGBaseT networks.

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

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

The program can be used on PCs of IBM PC architecture with Windows 10/11 OS.

The program was developed in the Delphi 10.4.1 environment.

Keywords: computer science, NGBaseT

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	6
1.1 Призначення системи.....	6
1.2 Область застосування.....	8
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	10
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	10
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	16
2.3 Розгорнута постановка завдання	21
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	23
3.1 Опис функціонування системи	23
3.2 Розробка структурної схеми.....	28
3.3 Розробка функціональної схеми	30
3.4 Розробка діаграми процесів.....	33
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	35
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	35
4.2 Захист розробленого програмного забезпечення.....	64
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	66
6 НАУКОВА НОВИЗНА	69

					ВКРМ-122.23.0079.00.00.ПЗ			
Вим	Арк.	№ докум.	Підп.	Дата	<i>Дослідження та програмна реалізація системи моніторингу центрів обробки даних, побудованих з використанням мереж NGBaseT</i>	Літ.	Аркуш	Аркушів
<i>Розроб.</i>	<i>Маляренко Р.В.</i>					М	1	110
<i>Перев.</i>	<i>Писенко І.А</i>					ЦНТУ КН-22МЗ		
<i>Н.контр.</i>	<i>Коваленко А.С.</i>							
<i>Затв.</i>	<i>Смірнов О.А.</i>							

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	70
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	70
7.2 Розрахунок трудомісткості розробки програмної продукції.....	72
7.3 Визначення чисельності виконавців і планового фонду зарплати.....	74
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника.....	79
7.5 Визначення собівартості розробки та ціни програмної продукції.....	83
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	86
7.7 Визначення експлуатаційних витрат.....	86
7.8 Визначення економічної ефективності програмної продукції.....	88
7.9 Висновок.....	90
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	91
8.1 Вступ.....	91
8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером.....	92
8.3 Розробка заходів з умов поліпшення охорони праці.....	93
8.4 Пожежна безпека.....	94
8.5 Розрахункова частина	97
9 ОСНОВНІ ВИСНОВКИ.....	101
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	103

ВСТУП

Актуальність теми. Технічний білий документ від CommScore надає детальну інформацію про останні розробки IEEE для NGBASE-T і відповідних специфікацій кабелів в Асоціації телекомунікаційної промисловості (TIA) у Сполучених Штатах (США) і в Міжнародній організації стандартизації (ISO)/Міжнародний електротехнічний комітет (IEC) на підтримку цієї програми.

Як зазначено в документі, «NGBASE-T означає наступне покоління BASE-T, додаток Ethernet, що перевищує 10 Гбіт/с через виту пару або збалансований кабель. Інститут інженерів з електротехніки та електроніки (IEEE) 802.3 NGBASE-T Call for-Interest (CFI), що призвело до формування дослідницької групи для дослідження та, можливо, розробки цієї технології. Основним застосуванням цієї технології є середовище центру обробки даних для з'єднань між серверами та комутаторами в межах ряду, як правило, називають архітектурою кінця рядка або середини рядка».

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи моніторингу центрів обробки даних, побудованих з використанням мереж NGBaseT.

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

– Огляд існуючих систем моніторингу центрів обробки даних, побудованих з використанням мереж NGBaseT.

– Дослідження системи моніторингу центрів обробки даних, побудованих з використанням мереж NGBaseT.

– Програмна реалізація системи моніторингу центрів обробки даних, побудованих з використанням мереж NGBaseT.

Об'єктом дослідження є процес моніторингу центрів обробки даних, побудованих з використанням мереж NGBaseT.

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

Предметом дослідження є методи моніторингу центрів обробки даних, побудованих з використанням мереж NGBaseT.

Методи дослідження базуються на методах хмарних технологій, методах математичної статистики, методах розробки програмного забезпечення.

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

– Удосконалено метод моніторингу центрів обробки даних, побудованих з використанням мереж NGBaseT.

– Розроблено вітчизняний продукт моніторингу центрів обробки даних, побудованих з використанням мереж NGBaseT, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі моніторингу центрів обробки даних, побудованих з використанням мереж NGBaseT.

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

Робота апробована на LVII Науково-технічній конференції здобувачів вищої освіти «Наука – виробництву», 2023, основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у роботі збірника праць молодих науковців ЦНТУ, випуск №14.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи моніторингу центрів обробки даних, побудованих з використанням мереж NGBaseT, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Система призначена для моніторингу центрів обробки даних, побудованих з використанням мереж 10GBaseT. Нова технічна документація від **Broadcom** ілюструє, як кабелі 10GBase-T і вита пара можуть значно знизити капітальні витрати на з'єднання в центрі обробки даних. У документі висвітлено способи, за допомогою яких з'єднання центру обробки даних включає як з'єднання між комутатором Top-of-Rack (ToR) і сервером, так і з'єднання між ToR і комутатором.

Аналіз показує, як 10GBase-T знижує витрати шляхом заміни дорогих кабелів SFP+ DAC на доступні UTP-кабелі та завдяки тому, що архітектори дають змогу оптимізувати фізичні топології.

Щойно після створення Комітету центру обробки даних Асоціація комунікаційного кабелю та зв'язку (СССА) випустила технічний документ під назвою «Ознайомлення переваг і недоліків структурованих кабелів у порівнянні з верхньою частиною стійки в центрі обробки даних». Згідно з СССА, документ «вивчає багато факторів, які слід враховувати при оцінці верхньої частини стійки (ToR) і конфігурацій структурованих кабелів. Теми включають вплив цих конфігурацій на загальне управління; масштабованість і оновлення; сумісність; витрати на обладнання, технічне обслуговування та прокладання кабелів; використання порту; вимоги до енергоспоживання та охолодження». Асоціація каже, що її Комітет центрів обробки даних планує випустити більше офіційних документів.

Виконавчий директор СССА Френк Пері прокоментував: «Темпи, з якими змінюється апаратне забезпечення центру обробки даних і конфігурація простору, лякають. СССА визнала як потребу, так і можливість допомогти у

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

прийнятті рішень щодо кабелів центрів обробки даних, надаючи останні дослідження, варіанти та експертні думки від провідних виробників кабелів і підключення. Як і в інших наших робочих групах, мета Комітету Центру обробки даних СССА – додати наш голос до розробки галузевих кодексів, стандартів та інших важливих ресурсів».

Базуючись на змісті інформаційного документа ToR-vs.-structured-cabling white paper, цей голос закликає розглянути керованість центру обробки даних, включаючи операційну структуру та політику. Як зазначено в документі, у конфігурації ToR переміщення/додавання/зміни «можуть бути більш складними та трудомісткими, особливо у великих центрах обробки даних із сотнями шаф. Зміни повинні бути зроблені в окремих стійках або шафах, а не в одній зручній центральній зоні латання». Реалізація архітектури ToR, як пояснюється в документі, пряме підключення комутаторів у стійці до серверів у тій самій стійці усуває центральний простір для латання. У цих сценаріях ТЗ, де немає централізованого виправлення, «визначення конкретної стійки або шафи, які потребують змін, може бути складним процесом», – додає документ.

У роботі також розглядаються такі питання, як масштабованість і оновлення, сумісність, технічне обслуговування, використання портів комутатора, енергоспоживання, охолодження та інші.

Боб Карлсон з The Siemon Company, який очолює комітет центру обробки даних СССА, сказав, що комітет планує реалізувати активну, постійну глобальну комунікаційну програму з використанням різноманітних місць. «Середовище глобального центру обробки даних є динамічним і складним для тих, хто проектує кабельну мережу», – сказав Карлсон. «Вибір дизайну та топології кабельних систем значно впливає на використання серверів і портів, ефективність роботи та навіть споживання енергії. Новий комітет прагне надати інформацію та ідеї, актуальні в усьому світі, щоб допомогти професіоналам у сфері дизайну та кінцевим користувачам приймати обґрунтовані рішення щодо кабельної розводки».

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

1.2 Область застосування

Областю застосування системи є ЦОД. Центр Обробки Даних (ЦОД) – високотехнологічне спорудження для розміщення обчислювального устаткування. Споконвічно ЦОД використовувалися, в основному, для власних потреб підприємств і організацій. Останнім часом даний термін одержав широке поширення в комерційній сфері, через зростання інтересу до послуг часток ЦОД і попиту на них серед клієнтів.

Послуги ЦОД

Центри обробки даних пропонують своїм клієнтам цілий комплекс телекомунікаційних послуг, пов'язаних зі зберіганням і обробкою інформації. Крім стандартних рішень, деякі ДЦ пропонують додаткові сервіси.

Стандартні сервіси центрів обробки даних:

- Оренда стійка.
- Оренда серверів.
- Colocation.

Додаткові сервіси центрів обробки даних:

- Бекап.
- Хмарні рішення.
- Адмініструємий сервер.
- Віддалений робочий стіл.

Технології, застосовувані в ЦОД

Високорозвинена технічна інфраструктура, що дозволяє підтримувати оптимальні умови для клієнтського встаткування – ключова характеристика для сучасного дати-центра.

Центри обробки даних звичайно мають декілька інтернет-каналів від різних провайдерів для забезпечення безперебійної роботи власного й клієнтського встаткування.

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

Відмінності центрів обробки даних

Стандарт ТІА-942

По цьому стандарті всі ДЦ одержують певний рівень – від tier1 до tier4.

Формат

За цим критерієм виділяють:

- автономні центри обробки даних;
- дата-центри, що орендують площі в більших провайдерів;
- серверні, що звичайно використовуються під потреби конкретного підприємства;
- «треш» дата-центри (не мають належної інфраструктури).

Таким чином, виходячи з вищеперахованого, дослідження та програмна реалізація системи моніторингу центрів обробки даних, побудованих з використанням мереж NGBaseT, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

КБПЗ-2023

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

Ключовим нововведенням є модуль інвентаризації, завдяки якому з'явилася можливість автоматично здійснювати збір інформації про програмне й апаратне забезпечення комп'ютерів у мережі й вести облік її змін. Додаток дозволяє відображати докладну інформацію з одного пристрою або зведену інформацію із групи пристроїв. Наприклад, можна подивитися список різних моделей принтерів в організації, довідатися, у кого вони встановлені. Окремо відображається зведена інформація з одного комп'ютера й детальна інформація з кожного з його комплектуючих.

iTMan24

Компанія iTMan представила нову версію свого рішення для обліку й моніторингу IT-інфраструктури iTMan24 для установки в приватній хмарі.

Версія призначена для компаній, які через правові обмеження не можуть користуватися публічними «хмарами». iTMan24 Private Cloud має той же функціонал, що й публічний сервіс. До числа ключових можливостей рішення ставляться інвентаризація комп'ютерів у мережі організації, інвентаризація програмного забезпечення, додавання й облік ліцензій на ПЗ, моніторинг використання співробітниками додатків на ПК. Рішення iTMan24 Private Cloud орієнтоване на великі компанії й інші організації, яким у силу специфіки діяльності, особливостей IT-інфраструктури й правових обмежень не підходить стандартна реалізація сервісу iTMan24.

Якщо вас уже дістало те, що комп'ютери у вас в офісі постійно ламаються, програми не працюють, а ваш адміністратор займається рішенням постійних проблем з мережею і нічого не встигає, то самий час взяти IT інфраструктуру під контроль. Для цього існують спеціальні програми й сервіси. Але якщо ви почнете копати в цьому напрямку – вас завантажать незрозумілими словами ITSM, ITIL, кращими практиками й такими ж незрозумілими інструментами, розібратися з якими без пляшки не можна. Ви встановлюєте спеціальну програму-агент на всі ваші комп'ютери. Агент збирає дані про комп'ютер, установлених програмах і їхньому використанні й передає їх в iTMan24. Ви заходите у свій онлайн кабінет з

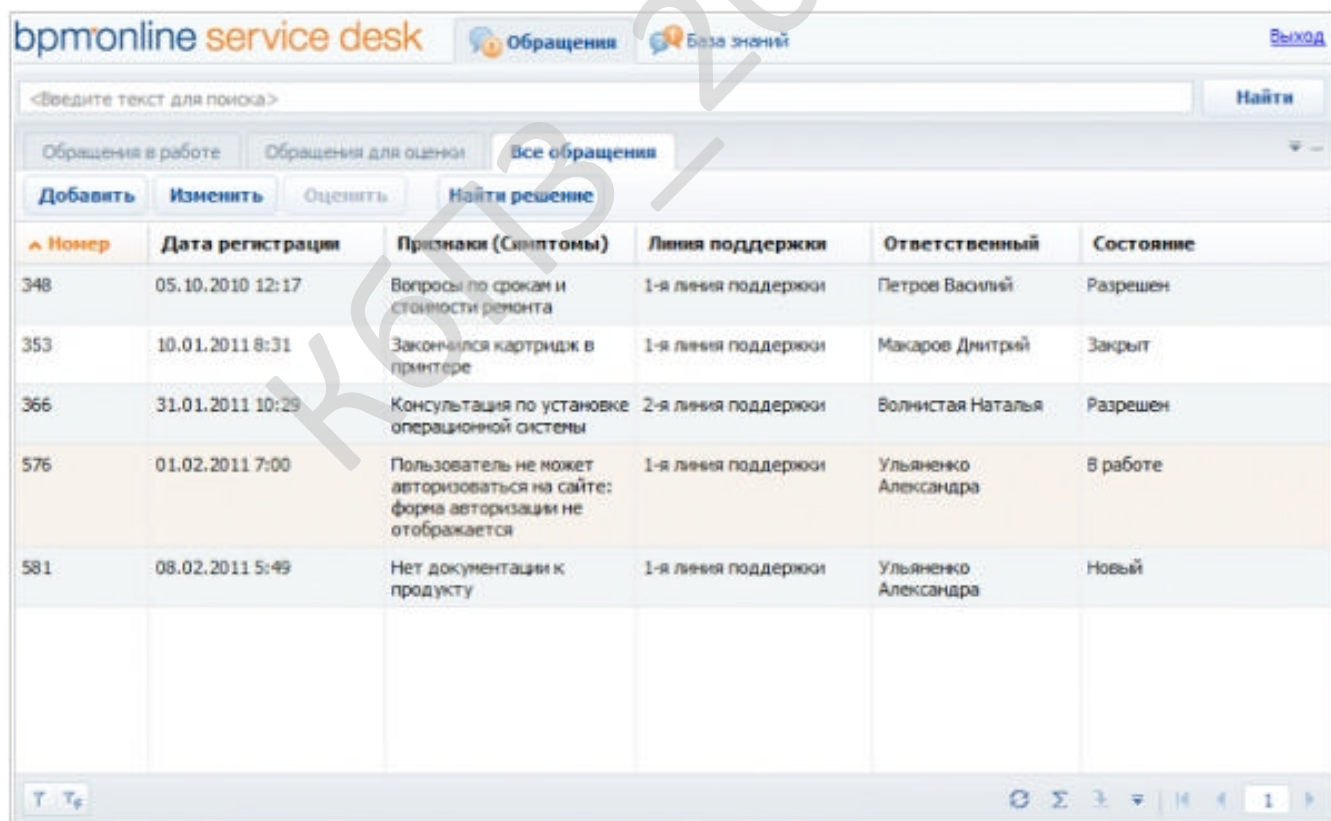
					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

комп'ютера або смартфона й бачите: що працює, що не працює, що потрібно оновити або ліцензувати.

BPMonline Service Desk

Компанія Terrasoft представила нову версію системи BPMonline Service Desk. У ній значно розширилися можливості роботи з електронною поштою (включаючи налаштування email повідомлень і автоматичну реєстрацію email), з'явилися можливості відстеження «здоров'я» сервісів, автоматичної реєстрації інцидентів (завдяки убудованій інтеграції із системами моніторингу), розширені можливості взаємодії системи з каталогом LDAP, доданий механізм зручного керування паспортами й моделями конфігураційних одиниць.

Крім того, з'явилася редакція Team Edition для невеликих сервісних служб (до 5 співробітників), що коштує \$1500 євро в рік. Ця редакція дозволяє автоматизувати базові процеси ITIL і містить у собі клієнтський портал.



The screenshot displays the BPMonline Service Desk interface. At the top, there is a search bar with the text '<Введите текст для поиска>' and a 'Найти' button. Below the search bar, there are tabs for 'Обращения в работе', 'Обращения для оценки', and 'Все обращения'. A row of action buttons includes 'Добавить', 'Изменить', 'Оценить', and 'Найти решение'. The main area contains a table with the following columns: '№', 'Дата регистрации', 'Признаки (Симптомы)', 'Линия поддержки', 'Ответственный', and 'Состояние'. The table lists five tickets with their respective details.

№	Дата регистрации	Признаки (Симптомы)	Линия поддержки	Ответственный	Состояние
348	05.10.2010 12:17	Вопросы по срокам и стоимости ремонта	1-я линия поддержки	Петров Василий	Разрешен
353	10.01.2011 8:31	Закончился картридж в принтере	1-я линия поддержки	Макаров Дмитрий	Закрыт
366	31.01.2011 10:29	Консультация по установке операционной системы	2-я линия поддержки	Волнистая Наталья	Разрешен
576	01.02.2011 7:00	Пользователь не может авторизоваться на сайте: форма авторизации не отображается	1-я линия поддержки	Ульяненко Александра	В работе
581	08.02.2011 5:49	Нет документации к продукту	1-я линия поддержки	Ульяненко Александра	Новый

Рисунок 2.2 – Интерфейс користувача BPMonline Service Desk

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

ITSM 365

Компанія Naumen – це лідер вітчизняного ринку ITSM систем (систем для керування IT інфраструктурою). Але в основному її клієнтами є великі компанії – банки, телекоми й федеральні торговельні мережі. Воно й зрозуміло. Naumen Service Desk – це дуже навірена й дорога платформа. На початку року Naumen запустив SaaS-Версію Naumen Service Desk. Це та ж сама навірена система, але використовувана із хмари (на умовах оренди), тому за вартістю вона знову ж не стала доступної для SMB. І от нарешті-те, Naumen представив SaaS сервіс для малих-середніх компаній – ITSM 365. Цей сервіс являє собою вже не складну платформу, а готове рішення із передналаштованим інтерфейсом, логікою, процесами для малого-середнього бізнесу.

Kaseya Traverse

Компанія Kaseya, розроблювач ITSM систем, оголосила про випуск рішення Kaseya Traverse, що надається по моделі SaaS і забезпечує моніторинг корпоративної мережі для хмарних, віртуалізованих і розподілених середовищ. Traverse синтезує й автоматично надає необхідну інформації про інфраструктуру, дозволяючи рішенню почати працювати негайно. Воно розпізнає й запам'ятовує нормальну модель поведження мережі залежно від часу доби, дня тижня й місяця або пори року й автоматично встановлює різні пороги на основі цих моделей. Тому про аномалію в поведженні систем рішення відразу повідомляє IT-фахівцям. Traverse також забезпечує убудовану мережну конфігурацію керування, що дозволяє створювати резервні копії й відновлювати конфігурацію всіх керованих мережних пристроїв (межмережні екрани, комутатори, маршрутизатори й т.д.) і розсилає повідомлення при виявленні змін у конфігурації.

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

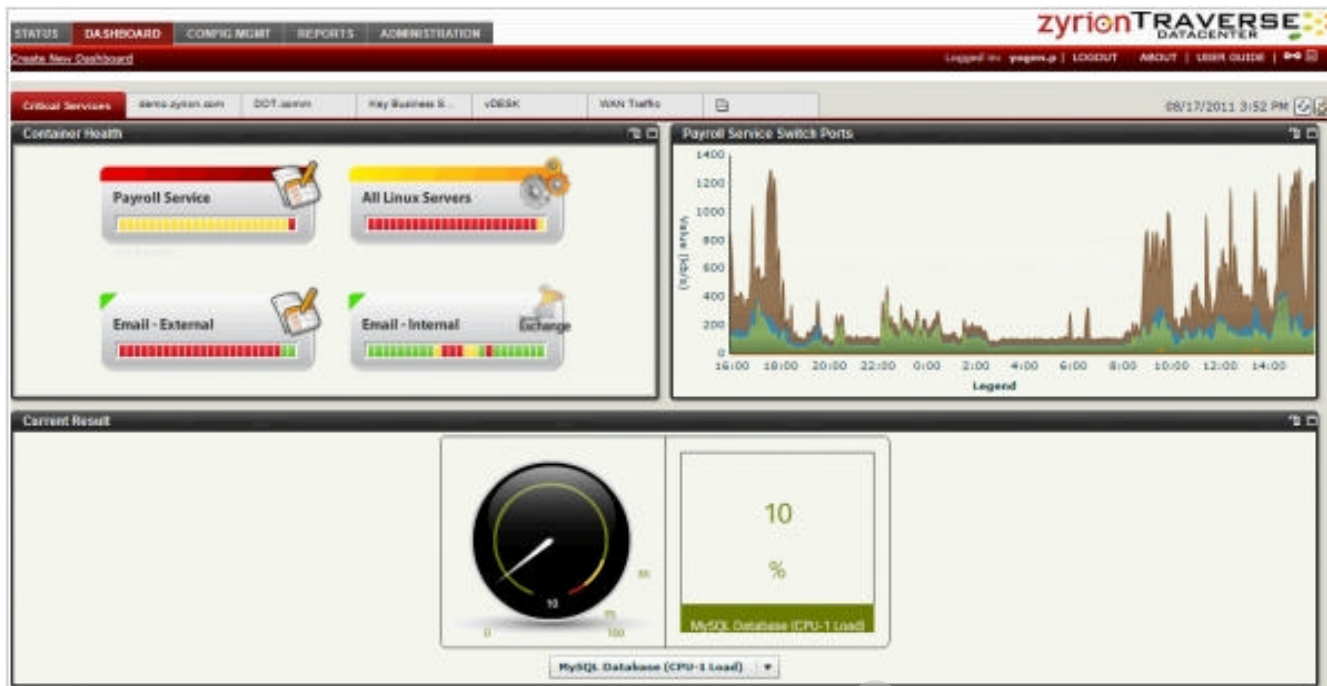


Рисунок 2.3 – Інтерфейс користувача Kaseya Traverse

Smartnut

Компанії ProLAN і Smartnut випустили спільне рішення на базі своїх продуктів: онлайн Service Desk системи Smartnut, системи моніторингу IT-інфраструктури ProLAN SLA-ON і системи швидкої діагностики інцидентів Червона Кнопка. У рамках рішення реалізована інтеграція Smartnut і Червоної Кнопки з метою автоматичної реєстрації інцидентів із прикріпленням всієї необхідної для їхнього рішення інформацією. У випадку виникнення проблеми користувачеві досить просто натиснути на спеціальне USB-пристрій у своєму комп'ютері, і тикет з усією діагностичною інформацією з'явиться в Service Desk. Комплексне рішення можна придбати із знижкою 15% (стосовно ціни при покупці окремих продуктів).

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

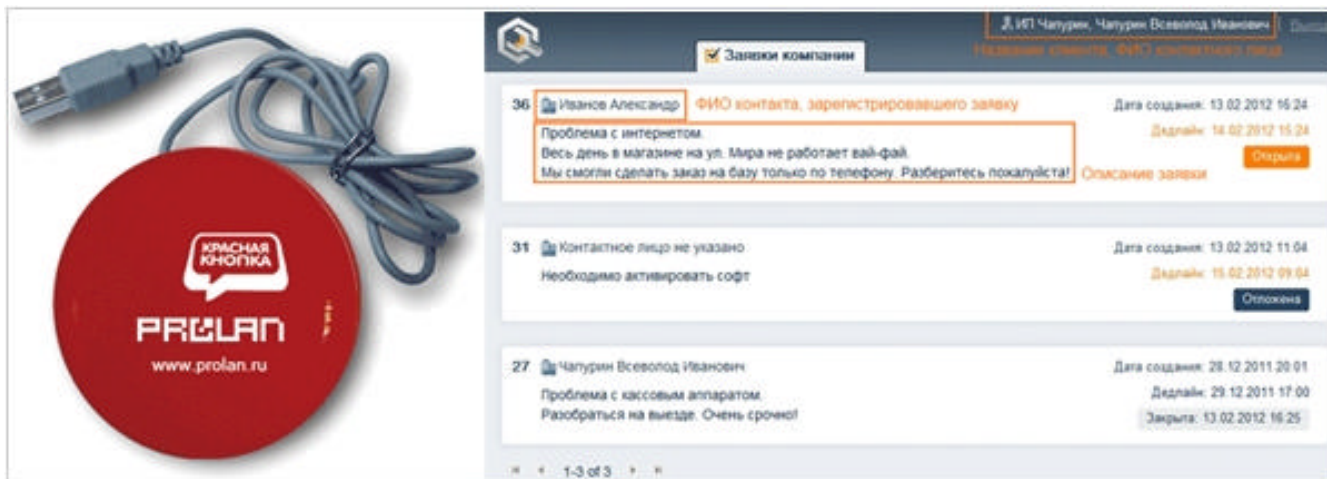


Рисунок 2.4 – Інтерфейс користувача Smartnut

IBM Tivoli Live

IBM представила новий SaaS сервіс IBM Tivoli Live Monitoring Services – рішення, що дозволяє моніторити доступність джерел даних, операційних систем, систем віртуалізації й бізнесів-додатків. Сервіс, по ідеї, повинен попередити IT-менеджера перш ніж почнуться серйозні проблеми у функціонуванні IT інфраструктури. Tivoli Live працює за допомогою інстальованого агента IBM Tivoli Monitoring, що збирає дані в на локальному сервері й відправляє їх у хмару. Сервіс Tivoli Live хоститься в дата-центрі IBM. Нагадаємо, що на початку місяця IBM також випустила хмарне рішення Tivoli для Amazon Web Services, що відмінно підходить тим компаніям, які розміщують елементи своєї IT-інфраструктури на AWS.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Embarcadero Delphi, раніше Borland Delphi і Codegear Delphi, – інтегроване середовище розробки ПЗ для Microsoft Windows, Mac OS, iOS і Android мовою Delphi (що раніше носила назву Object Pascal), створена спочатку фірмою Borland і на даний момент приналежна й розроблювальна Embarcadero Technologies.

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

Embarcadero Delphi є частиною пакета Embarcadero RAD Studio і поставляється в чотирьох редакціях: Community (поширюється безкоштовно й має обмежену ліцензію на використання в комерційних цілях), Professional, Enterprise і Architect.

Delphi 10.4 Sydney

Випущено 26 травня 2020 року. RAD Studio Delphi 10.4 забезпечує значно поліпшену високопродуктивну нативну підтримку Windows, кращу продуктивність розробки, миттєві підказки code completion, прискорення виконання коду із синтаксисом керованих записів, поліпшення виконання паралельних завдань на сучасних багатоядерних CPU, а також містить більш 1000 виправлень багів, поліпшення продуктивності середовища й бібліотек і багато чого крім того.

Основні можливості Delphi 10.4.1:

– Істотні розширення для Windows: поліпшення для застосунків на моніторах 4K High DPI, інтеграція з новим WebView2 на базі Chromium, використання розширених title bars, таких же, як в Office, Explorer, Google Chrome.

– Керування пам'яттю в Delphi тепер стандартизоване на всіх підтримуваних платформах – мобільних, настільних і серверних – використовувачи класичну реалізацію керування пам'яттю об'єктів.

– Істотне поліпшення Delphi Code Insight (без можливого блокування IDE – в окремому процесі), що допоможе при роботі з великими проектами.

– Тип даних Delphi «record» тепер підтримуватимуть довільні ініціалізацію, фіналізацію й операції копіювання.

– Розширена підтримка бібліотек C++: ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode.

– Відладник Win 64 (на LLDB) і збирач для C++.

– Поліпшення для C++: Включена велика кількість поліпшень STL з Dinkumware.

– Підтримка Metal Driver GPU для macOS і iOS.

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

– Вбудований Fmxlinux.

– Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.

Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

Реалізований заново стилізуємий FMX компонент TМемо на платформі Windows значно поліпшений і тепер має відмінну підтримку ІМЕ.

– Численні поліпшення швидкості й стабільності роботи нашої бібліотеки The Parallel Programming Library (PPL).

– Додані оновлені драйвери для FireBird, PostgreSQL і SQLite.

– Клієнтські бібліотеки HTTP і REST Client розширені застосунковими можливостями роботи з HTTPS. Також були розширені можливості підтримки Amazon AWS services

– У технологію Visual LiveBindings внесена безліч поліпшень, у тому числі швидкодії, що стосуються, застосунків на VCL і FireMonkey

RAD Studio 10.4 Короткий огляд:

– Істотні розширення для Windows. Створення застосунків, що чудово виглядають, із чіткими елементами інтерфейсу на 4k моніторах High DPI за допомогою нової гнучкої підтримки стилів елементів керування на екрані. Інтеграція із сучасними, безпечними web-технологіями від Microsoft – новим WebView2 на базі Chromium. Використання сучасних розширених title bars, таких же, як в Office, Explorer, Google Chrome, у своїх проектах. Істотні поліпшення надійності налагодження в новому відладнику для C++ Windows 64-bit.

– Зросла продуктивність розробки. Ріст продуктивності за рахунок миттєвої реакції підказок code completion у середовищі IDE. Краща сумісність із уже наявною кодовою базою, і спрощення програмування за рахунок уніфікованої архітектури керування пам'яттю. Швидке зв'язування даних і візуальних елементів за допомогою розширеної технології Visual LiveBindings з підвищеною швидкодією. Просте використання розповсюджених бібліотек C++, наприклад, ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode. Оновлена підтримка Amazon AWS cloud.

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

– Поліпшення швидкодії і якості. Більш 1000 поліпшень швидкодії і якості. Краща ефективність коду за допомогою нового синтаксису `custom managed records`. Більш швидке виконання паралельних завдань на сучасних багатоядерних CPU. Переконаєтеся в прискоренні відображення на екрані з підтримкою Metal API на macOS і iOS. Краща сумісність із уже наявною кодовою базою й спрощення програмування за рахунок уніфікованої архітектури керування пам'яттю.

Істотне поліпшення Delphi Code Insight

Як найбільше й головне поліпшення інструментів програмування Delphi за багато років, в 10.4 Delphi Code Insight реалізований через Language Server Protocol (LSP). LSP – це технологія генерації результатів для code completion, навігації й інших сервісів в окремому процесі. Це значить, що code completion і Code Insight одержать більш точні результати без блокування IDE. 10.4 забезпечує набагато більш високу продуктивність розроблювачів, які працюють із більшими проектами, що містять мільйони рядків коду.

Delphi Custom Managed Records

Ключове розширення мови Delphi: тип даних Delphi «record» тепер підтримуть довільні ініціалізацію, фіналізацію й операції копіювання. Управляйте тем, як ці структури створюються, копіюються й звільняються з допомогу вашого коду, який буде виконуватися у відповідний момент.

Це розширює потужність конструкцій records в Delphi, які використовуються щоб одержати більшу ефективність у порівнянні із класами.

Єдине керування пам'яттю

Керування пам'яттю в Delphi тепер стандартизоване на всіх підтримуваних платформах – мобільних, настільних і серверних – використовувачи класичну реалізацію керування пам'яттю об'єктів.

У порівнянні з Automatic Reference Counting (ARC), це дає кращу сумісність із існуючим кодом і спрощує написання компонентів, бібліотек і застосунків.

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

ARC модель керування пам'яттю model залишилася для керування рядками й посиланнями на тип інтерфейсу на всіх платформах. Для C++ це означає, що при створенні й звільненні Delphi-style класів в C++ використовується звичайне керування пам'яттю, як у будь-якого heap-allocated класу C++, що значно знижує складність коду.

Розширена підтримка бібліотек C++

В 10.4 ми портували багато популярних бібліотек C++ у C++Builder.

Забезпечивши оптимізовану підтримку бібліотек ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode, поряд із уже підтримуваними Boost і Eigen, які можуть бути додані за допомогою менеджера пакетів Getit.

Win 64-відладник і збирач для C++

В 10.4 з'явився новий відладник C++ для Windows 64-bit. Відладник заснований на LLDB і показує значне збільшення стабільності при налагодженні 64-bit застосунків поряд з новими відладочними можливостями, такими як перегляд і інспекція типів начебто рядків C++ і Delphi, а також колекцій STL, включаючи std::vector, std::map і інших. Крім того, згенерована для застосунку відладочна інформація має інший внутрішній формат, сприяючи більш стабільному й багатому на можливості процесу налагодження, більш докладним перегляду й інспекції в debug-time.

Підвищення якості й швидкодії інструментів

- Велика кількість поліпшень STL від Dinkumware.
- Поліпшені деякі найважливіші методи й області RTL, на базі поліпшень сумісності з популярними бібліотеками C++.
- Поліпшена підтримка Snake.
- Велика кількість виправлень для підвищення стабільності і якості.
- Відновлення Windows API – Обновлено й додали безліч декларацій API щоб добитися ще більшої інтеграції із платформою Windows.

– Загальні вдосконалення в бібліотеці доступу до БД FireDAC, включаючи оновлені драйвера для FireBird, PostgreSQL і SQLite. Вибір статичного або динамічного підключення SQLite до застосунку.

Змінені стилі VCL для High DPI

В 10.4, архітектура стилізації VCL була суттєво розширена для підтримки High DPI і 4K моніторів. Тепер усі елементи UI на формі VCL автоматично масштабуються під відповідне до монітора дозвіл для показу форми. Був оновлений API стилізації для підтримки стилів high DPI.

Кожний графічний елемент UI може бути обраний з наборів різних масштабів і масштабований до потрібного DPI, що дає чітке зображення елементів UI на всіх моніторах.

Нові High DPI стилі й стилізація окремих VCL компонент

Обновлено велике число вбудованих і преміальних VCL стилів для підтримки нового режиму стилізації High-dpi. Це дозволяє вам створювати застосунку з відмінним дизайном для всіх моніторів.

Розроблювачі VCL застосунків тепер можуть використовувати трохи VCL стилів на різних формах в одному застосунку або в різних компонентах на одній формі. Це також включає стилізацію компонентів загальною темою для платформи. Крім застосункової гнучкості використання стилів, це дозволяє використовувати нестилізуємі компоненти із зовнішніх бібліотек в VCL застосунках, що використовують стиль.

Поліпшена кроссплатформеність

- Додана підтримка Metal Driver GPU для macOS і iOS.
- Крім підтримки останнього iOS SDK, в RAD Studio 10.4 розроблювачі можуть задовольнити нові вимоги Apple до набору стартових екранів.
- Реалізований заново стилізуємі FMX компонент TМемо на платформі Windows значно поліпшений і тепер має відмінну підтримку IME.
- Користувачам редакцій Enterprise або Architect доступна повна інтеграція Fmxlinux з IDE для створення клієнтських застосунків Linux з GUI.

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

- Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.
- Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

Оновлений менеджер пакетів Getit

Менеджер пакетів Getit в IDE був значно вдосконалений.

Дати випуску релізів пакетів тепер видні, і можливе сортування списку по цих датах; відбір тільки встановлених пакетів, контенту, доступного тільки при наявності підписки, багато чого іншого.

Універсальний інсталятор для установки Online і Offline

В 10.4 включений новий універсальний інсталятор, який використовує технологію на базі Getit. Цей інсталятор підтримує як online, так і offline (з ISO) варіанти установки.

Тепер обоє варіанта установки дозволяють вам указати початковий набір можливостей RAD Studio для установки, наприклад, свою комбінацію мов програмування й цільових платформ, мов інтерфейсу, і додавати до нього або видаляти непотрібне в будь-який момент.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випуск кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи моніторингу центрів обробки даних, побудованих з використанням мереж NGBaseT.

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

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

б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі.

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

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.

КБПЗ-2023

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Розглянемо технологію NGBaseT, на основі якої будуються центри обробки даних. Сервери й мережне встаткування з підтримкою технології 40 Gigabit Ethernet уже присутні на ринку, тому надзвичайно важливим стає проектування мереж передачі, які в найближчому майбутньому дозволять перейти на такі швидкості. Комітети зі стандартизації IEEE, TIA, ISO/IEC розробляють вимоги до електричних і фізичних характеристик трактів, які дозволять використовувати кручену пару як середовище передачі для 40Gb.

Ethernet є найпоширенішою технологією організації фізичного рівня мережі передачі даних. Головна причина її успіху полягає в тому, що, незважаючи на експонентний ріст обсягів переданих даних, вартість передачі (у перерахуванні на біт) зберігається на розумному рівні. Крім цього, популярності Ethernet сприяли низька вартість і простота впровадження, легкість в обслуговуванні й широкі можливості вдосконалення. І хоча швидкість передачі даних за три десятиліття існування Ethernet зросла на кілька порядків, швидкий розвиток таких сервісів і додатків, як стільникові мережі 4G, потокове відео, хмарні сервіси, вилучене зберігання й відновлення даних і інші, висуває надзвичайно високі вимоги до швидкості передачі даних, особливо в межах ЦОД.

Як середовище передачі даних для Ethernet використовуються кабелі на основі оптичного волокна або кручених пар. Спочатку при освоєнні нових рубежів швидкості Ethernet реалізовувалася по волоконній оптиці, однак через деякий час аналогічні швидкості виявлялися досяжні й із застосуванням крученопарного середовища передачі. Укупі з більше низькою вартістю розгортання це прискорювало широке поширення чергового більше швидкісного варіанта технології.

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

У цей час кабельні системи на основі кручених пар найбільше широко поширені в локальних мережах, особливо на рівні горизонтального розподілу сервісів Ethernet. Але й у ЦОД кручена пара застосовується в значних обсягах, оскільки однаково придатна для різних типів архітектури, використовуваних при побудові центрів обробки даних (Top of Rack, Middle of Row, End of Row).

Максимальна швидкість передачі даних, забезпечувана крученопарною проводкою, становить поки 10 Гбіт/с. Відповідний стандарт, IEEE 802.3an, де визначаються параметри протоколу передачі Ethernet на 10 Гбіт/с (10GBase), був опублікований у червні 2006 року як підсумок трирічної роботи з розробки вимог до нового протоколу з урахуванням збереження сумісності з попередніми версіями Ethernet і існуючими топологіями мереж.

Мережні топології, які повинен був підтримувати новий протокол, розроблялися з обліком уже існуючих топологій, призначених для підтримки більше низьких швидкостей передачі, і були орієнтовані переважно на розгортання кабельних мереж в офісах, а не центрах обробки даних з їхніми специфічними особливостями. У результаті активне встаткування, призначене для передачі 10 Gigabit Ethernet, повинне було підтримувати тракт передачі довжиною до 100 м, що містить 4 з'єднувачі (як для екранованого, так і неекранованого варіанта). Ці вимоги накладали значні обмеження на конструкцію активного мережного встаткування (позначуваного звичайно як РНУ) і тим самим значно затримали вивід нової технології на ринок.

Робоча група, що розробляє стандарт IEEE 802.3bq і відповідальна за визначення вимог до Next Generation Base, виділила деякі аспекти, пов'язані з кабельною проводкою, оптимізація яких дозволить прискорити вивід на ринок нового протоколу передачі даних. Нижче розглянемо ці основні можливості, що відкривають шлях до поліпшень.

Топологія мережі розраховуючи на більше короткі лінії

NGBase (40GBase) розробляється головним чином для застосування в ЦОД, де відстані між активним устаткуванням (від комутатора до сервера або до

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

пристрою зберігання даних) значно менше, ніж в офісних мережах. Менша довжина трактів передачі означає допустимість більше високих значень для відношення сигнал-шум, що, у свою чергу, сприяє скороченню енергоспоживання активного встаткування. У цей час зниження цього показника є одним з найважливіших факторів, прийнятих в увагу замовниками.

Використання винятково екранованої проводки

Такий вибір дозволить зменшити вплив міжкабельних наведень (Alien Crosstalk – наведення, що виникають між прилеглими кабелями,). Оскільки вимоги до рівня міжкабельних наведень були одними з найважливіших для коректної роботи протоколу 10GBase, системи, де застосовувалася неекранована елементна база, виявлялися значно більше сприйнятливими до цього параметра. Сьогодні більшість виробників кабельних систем згодні з тим, що забезпечити дотримання необхідних характеристик і, відповідно, надійну роботу NGBase (40Gb) можна тільки при використанні екранованої елементної бази. У результаті ті вендори, які раніше лобіювали неекрановані кабельні системи для підтримки 10GBase, погодилися з необхідністю застосування винятково екранованого варіанта в новому стандарті.

Вироблення однакових вимог до крученопарної елементної бази з боку TIA і ISO/IEC

Обидві організації (американська і міжнародна) розробляють стандарти на кабельні системи відповідно до вимог робочої групи IEEE до передатних характеристик для протоколу 40GBase. Виробіток однакових вимог до передатних характеристик дозволить уникнути плутанини (як це було у випадку 10GBase), що відбувається через те, що вимоги до окремих параметрів з боку TIA відрізняються від вимог ISO/IEC.

Робочі групи розглядають два можливих сценарії:

– перший позначається як Class I або Категорія 8 (залежно від комітету зі стандартизації) і базується на елементній базі Категорії 6A;

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

– другий – як Class II і передбачає використання елементної бази Категорії 7A. Обидва сценарії припускають збільшення частотної межі до 2 ГГц.

Облік внутрішнього компонування активного встаткування при моделюванні каналу (Channel)

Облік впливу цих елементів (магнетики, розведення плат і т.п.) на модель кабельної системи на етапі визначення вимог повинен сприяти зменшенню перекручувань сигналу при передачі й, як наслідок, скороченню споживання енергії для його обробки.

З'єднувачі для 40GBase

Як і інші учасники ринку, фахівці TE Connectivity беруть активну участь у діяльності профільних робочих груп у складі комітетів зі стандартизації. Тим часом інженери компанії працюють над поліпшенням конструкції з'єднувача, в основу якого покладений добре відомий інтерфейс RJ45. Результатом цих зусиль стало досягнення вимог, заданих TIA Cat.8 (драфт 0.9 від 9 серпня 2013) і ISO/IEC Class I (драфт N2121 від 17 квітня 2013), при цьому забезпечується зворотна сумісність із існуючими сервісами Ethernet.

Передбачаючи потребу в з'єднувачах для високошвидкісної передачі даних, ще в 2007 році TE Connectivity розробила й випустила модульне гніздо AMP-TWIST 7AS SL Jack. Воно має частотну межу в 1000 МГц і має всі особливості, властиві продуктам серії AMP-TWIST. Це гніздо повністю відповідає твердим вимогам ISO/IEC Class II (драфт N2121 від 17 квітня 2013) у відношенні 40GBase.

Сервери й мережне встаткування з підтримкою технології 40 Gigabit Ethernet уже присутні на ринку, тому надзвичайно важливим стає проектування мереж передачі, які в найближчому майбутньому дозволять перейти на такі швидкості. Єдиним варіантом організації тракту передачі 40 Gigabit Ethernet довжиною більше 7 м зараз є застосування оптичного середовища передачі.

Комітети зі стандартизації (IEEE, TIA, ISO/IEC) розробляють вимоги до електричних і фізичних характеристик трактів, які дозволять використовувати

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

кручену пару як середовище передачі для 40 Gigabit Ethernet. Однак параметри, якими виробники активного встаткування можуть оперувати для розробки пристроїв, що підтримують протокол 40GBase, поки не специфіковані. Тому жоден проектувальник кабельної інфраструктури центра обробки даних не може бути впевнений у тім, що мідною проводкою буде підтримуватися швидкість 40 Гбіт/с, якщо тільки мова не йде про пряме з'єднання «точка-точка» (40GBaseCR4, до 7 м).

Крім того, не до кінця визначені вимоги відносно наступних аспектів:

- методик випробувань елементної бази на високих частотах, які потрібні для підтримки протоколу 40GBase;
- сумісності між сполучним і комутаційним устаткуванням різних виробників;
- методик польових випробувань змонтованої кабельної системи.

У той же час у результаті діяльності комітетів зі стандартизації досягнута згода у відношенні декількох ключових питань, що задає подальший напрямок роботи:

1. Основним інтерфейсом для активного встаткування (Media Device Interface, MDI) обраний RJ45 у силу його широкої поширеності й звичного форми-фактора. Інтерфейси інших типів теж можуть бути стандартизовані, якщо їхні робочі характеристики стануть перевершувати можливості RJ45 (див. також урізання «З'єднувачі для 40GBase»).

2. До робочих характеристик кабельної проводки повинні пред'являтися більше тверді вимоги, чим у випадку систем, що підтримують попереднє покоління Ethernet. Це дозволить спростити конструкцію приймачепередаючого встаткування й скоротити його енергоспоживання.

3. Для досягнення гарного рівня стійкості до електромагнітних перешкод необхідно використовувати екрановану елементну базу.

4. Імовірне розширення моделі тракту передачі сигналу на весь інтервал від приймачепередатчика до приймачепередатчика з метою скорочення

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

перекручувань сигналу при передачі й зменшення додаткової обробки (post-processing) даних.

У міру наближення дати публікації стандарту (можливо, це відбудеться в 2015 році) будуть точніше визначені електричні й фізичні параметри, і, як наслідок, постачальники кабельної продукції зможуть розробити на їхній основі ефективні рішення для створення телекомунікаційної інфраструктури, повністю відповідним новим вимогам. Поки занадто рано робити які-небудь висновки щодо параметрів протоколу 40GBase, тому необхідно ретельно аналізувати подібні заяви.

3.2 Розробка структурної схеми

Комплексна безпека центрів обробки даних, включає в себе захист від наступних погроз:

- відмова встаткування й програмного забезпечення;
- збої енергоживлення;
- пожежа й задимлення;
- несанкціонований доступ, злом, крадіжки;
- віруси;
- затоплення, різкі температурні зміни, пил;
- часткове руйнування будинку;
- електромагнітні випромінювання.

Переваги створення ЦОД:

– підвищення ефективності й надійності експлуатації обчислювальних ресурсів;

– надання відказостійких інфраструктурних сервісів у режимі 24 години x 7 днів у тиждень x 365 днів у році;

- просте й прозоре централізоване адміністрування;
- зниження витрат на надання інженерних комунікацій;

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

- високий рівень захисту інформаційної системи;
- централізоване керування й облік ресурсів ЦОД;
- контроль доступу до ЦОД;
- просте й зручне масштабування обчислювальних ресурсів.

Структурна схема системи зображена на рисунку 3.1.

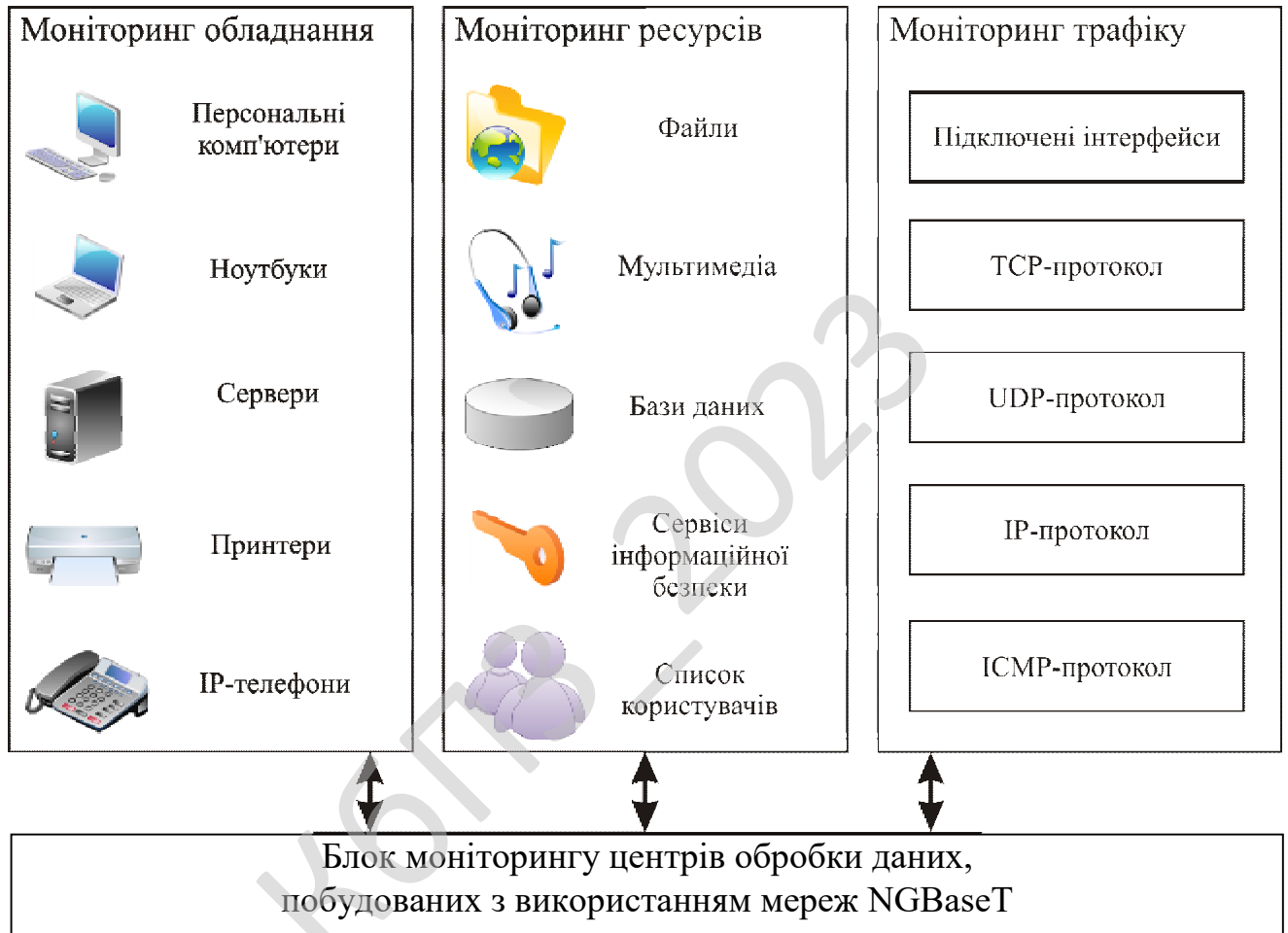


Рисунок 3.1 – Структурна схема системи

З рисунку видно, що моніторинг локальної мережі здійснюється з трьох напрямків:

- Моніторинг обладнання центрів обробки даних, побудованих з використанням мереж NGBaseT.
- Моніторинг ресурсів центрів обробки даних, побудованих з

використанням мереж NGBaseT.

– Моніторинг трафіку центрів обробки даних, побудованих з використанням мереж NGBaseT.

Моніторинг обладнання включає в себе побудову списку наявного обладнання та здійснення його контролю. До мережного обладнання, що підлягає моніторингу, відносяться: персональні комп'ютери, ноутбуки, сервери, принтери, ір-телефони.

Моніторинг ресурсів дозволяє переглядати та завантажувати наявні в мережі ресурси, а також розміщувати чи приховувати для загального доступу свої ресурси. До ресурсів локальної мережі відносяться: файли, мультимедіа, бази даних, сервіси інформаційної безпеки, список користувачів.

Моніторинг трафіку використовується для контролю вхідного та вихідного трафіку. Він включає у себе контроль підключених інтерфейсів, статистику подій по основним мережним протоколам: TCP, UDP, IP та ICMP.

3.3 Розробка функціональної схеми

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

– Моніторинг трафіку центрів обробки даних, побудованих з використанням мереж NGBaseT.

– Робота з файлами центрів обробки даних, побудованих з використанням мереж NGBaseT.

– Монітор з'єднань центрів обробки даних, побудованих з використанням мереж NGBaseT.

– Статистика подій центрів обробки даних, побудованих з використанням мереж NGBaseT.

– Робота з ресурсами центрів обробки даних, побудованих з використанням мереж NGBaseT.

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

– Робота з сесіями центрів обробки даних, побудованих з використанням мереж NGBaseT.

– Функції для роботи з центрами обробки даних, побудованих з використанням мереж NGBaseT.

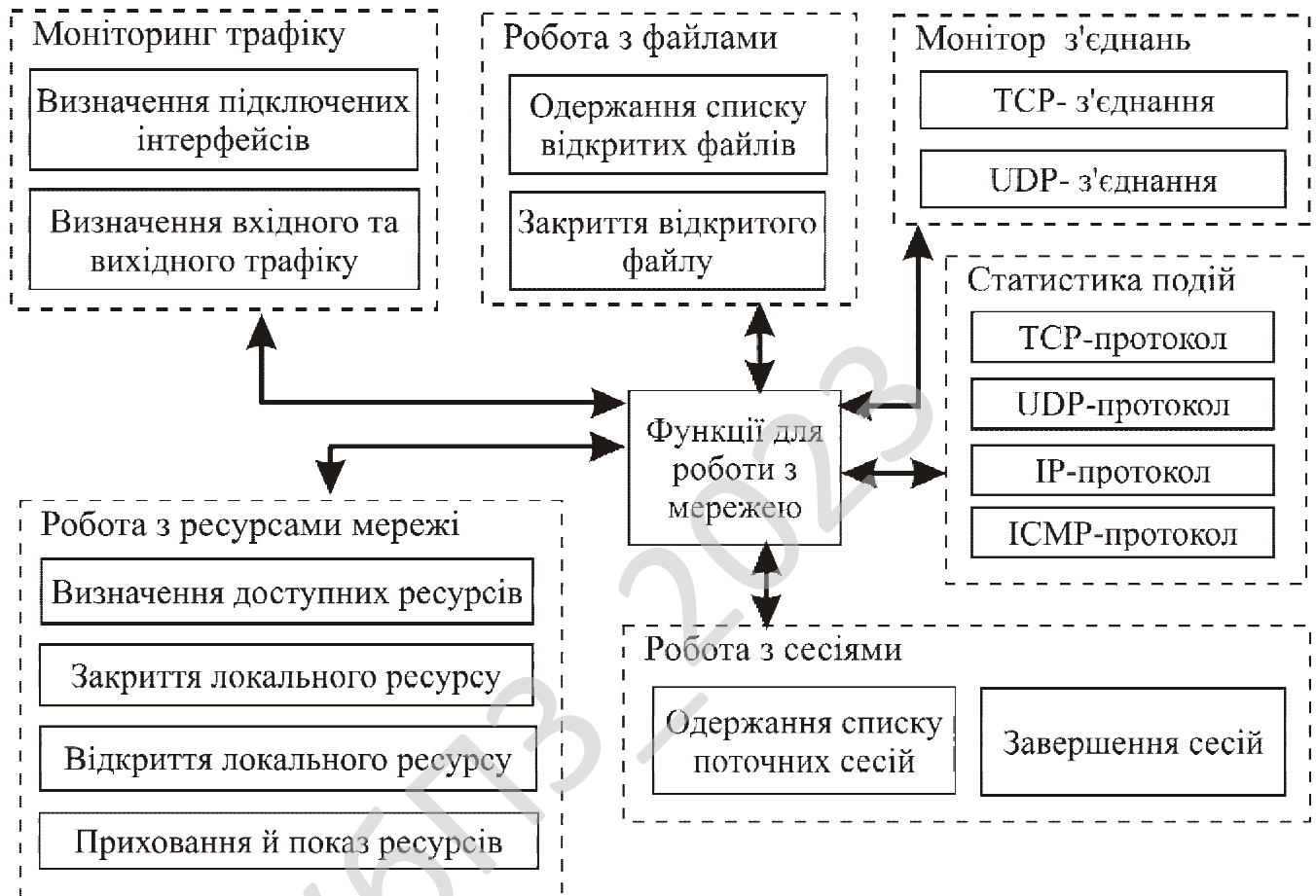


Рисунок 3.2 – Функціональна схема системи

Розглянемо детальніше кожний з блоків.

Моніторинг трафіку включає в себе:

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

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

Робота з файлами включає в себе:

- Одержання списку відкритих файлів.
- Закриття відкритого файлу.

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

Монітор з'єднань включає в себе:

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

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

Статистика подій включає в себе відстеження подій в наступних протоколах:

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

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

Робота з ресурсами мережі включає в себе:

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

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

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

Робота з сесіями включає в себе:

- Одержання списку поточних сесій.
- Завершення сесій.

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

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

3.4 Розробка діаграми процесів

Діаграма процесів розробленої системи зображена на рисунку 3.3. Після початку роботи розробленого ПЗ ми потрапляємо до головного блоку системи звідки через ланку дій відбувається наступне:

- Інтерфейс ПЗ.
- Статистика.
- Статистика по TCP-протоколу.
- Статистика по UDP-протоколу.
- Статистика по IP-протоколу.
- Статистика по ICMP-протоколу.
- Вивід на екран відкритих сесій.
- Закриття сесії.
- Відкриття сесії.
- Моніторинг системи.

- Пошук ресурсів локальної мережі.
- Побудова та виведення на екран дерева ресурсів.
- Пошук ресурсів по фільтру, вибраному користувачем.

Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі.

Діаграма взаємодії процесів використовується для візуалізації процесів обробки даних (структурне проектування).

Для розробника вважається звичним спочатку креслити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи.

Ця діаграма в подальшому підлягає уточненню шляхом деталізації процесів та потоків даних з метою показати систему що розробляється.

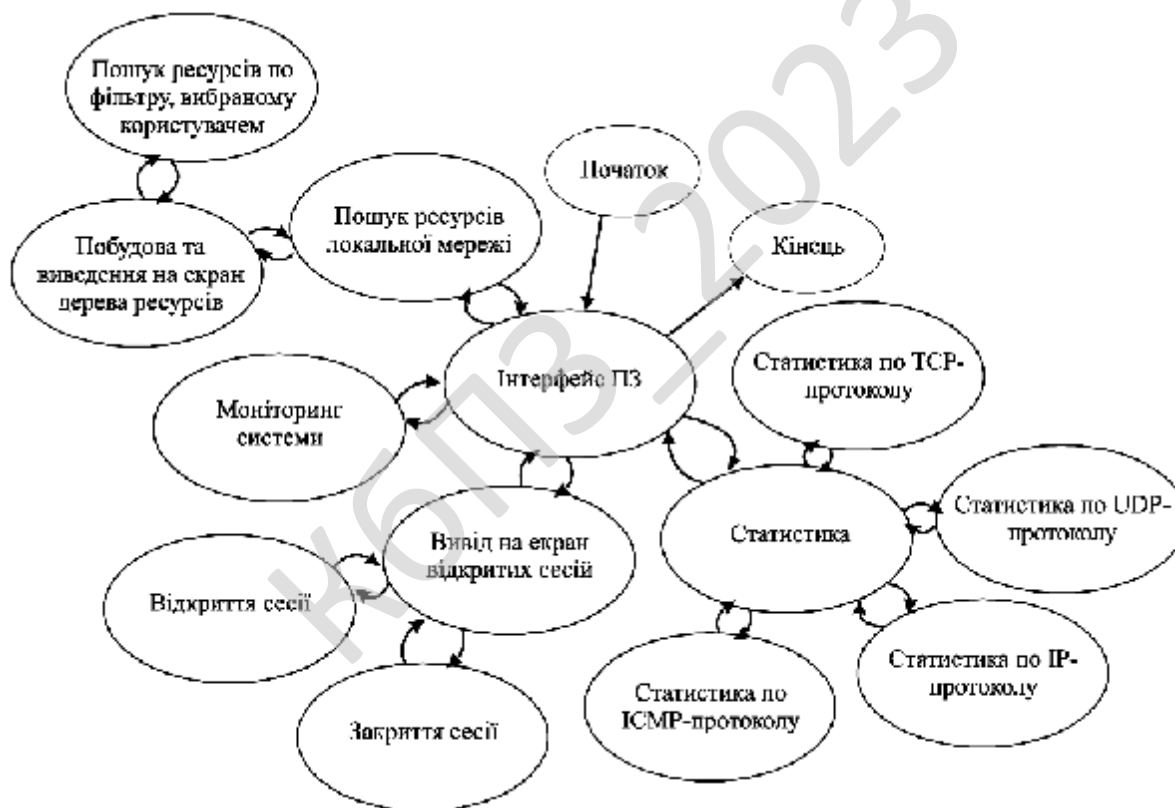


Рисунок 3.3 – Діаграма взаємодії процесів

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

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

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

На рисунку 4.1 зображена основна блок-схема програми, на рисунку 4.2 зображено роботу підпрограми. З 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 – перевірка поточного стану з завершенням роботи розробленого ПЗ.

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

Блок-схеми є першоджерелами стратегії розвитку ПЗ. Тому від точності і детальної блок-схеми залежить результат всієї програми. При виборі початкової точки відліку при побудові схем я враховував, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об'єктно-орієнтована що вимагає оптимізації програми високого рівня, також те, що при розробці програми слід надати особливу увагу модулю обробки поштових повідомлень.

При складанні блок-схем програмного забезпечення і напрацювання алгоритмів я зіткнувся з масою проблем, які вимагали напрацювання процедур і функцій над основною проблематикою.

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

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

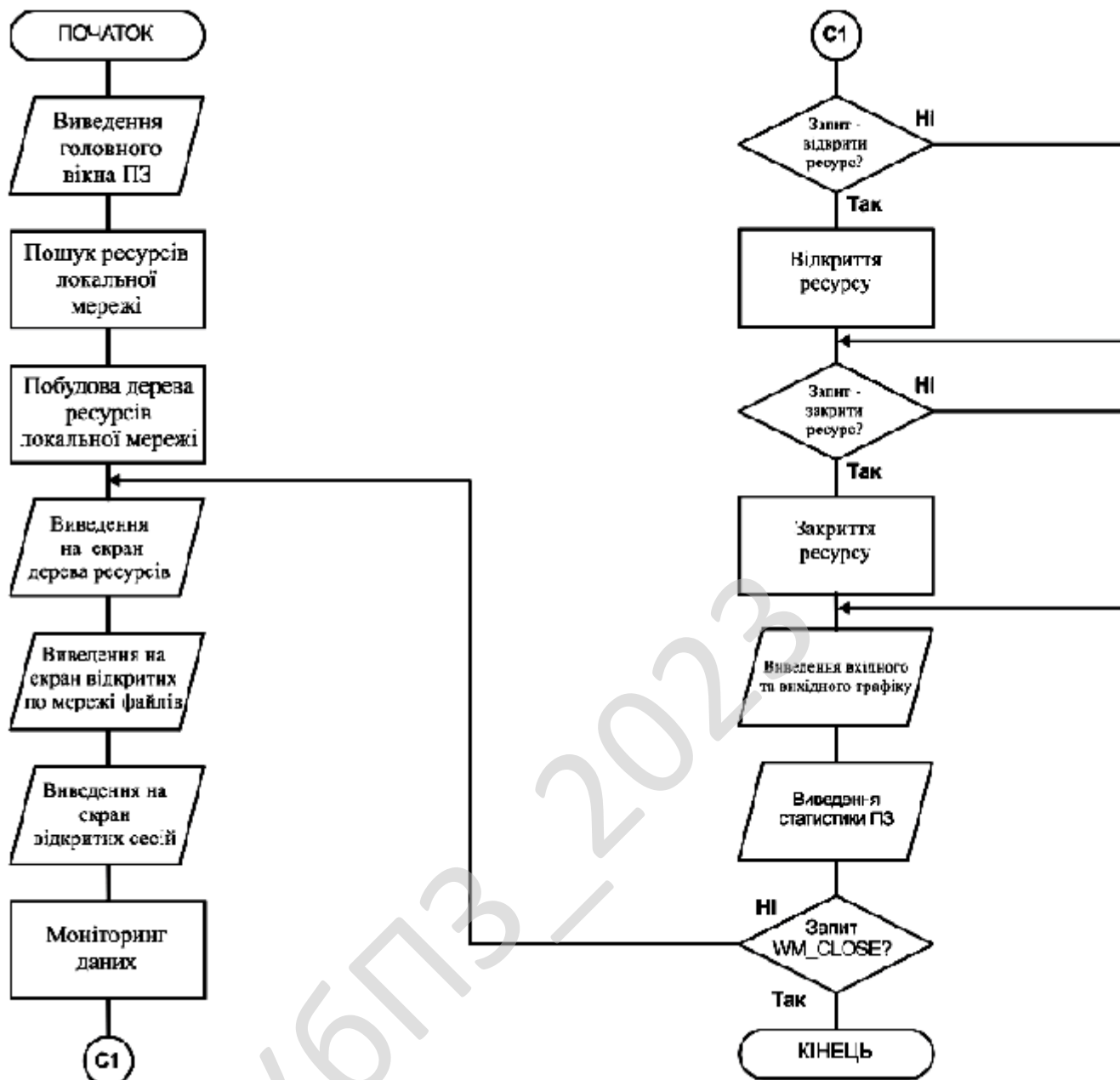


Рисунок 4.1 – Блок схема основної програми

Розглянемо функції, що надають інформацію про локальні ресурси й можливість їхнього контролю.

NetShareEnum – за допомогою цієї функції ми одержимо дані про всі свої й чужі загальні ресурси.

Оголошення функції для Windows 10/11:

```
var
    NetShareEnum :function (pszServer      : PChar;
```

```
sLevel      : Cardinal;  
pbBuffer    : PChar;  
cbBuffer    : Cardinal;  
pcEntriesRead,  
pcTotalAvail : Pointer):DWORD; stdcall;
```

Параметри:

- pszServer – повинен містити ім'я віддаленого комп'ютера, на якому повинна виконається функція, якщо виконуємо в себе то даному параметру можна привласнити NIL;
- sLevel – повинен містити ідентифікатор структури;
- pbBuffer – повинен містити покажчик на масив структур;
- cbBuffer – повинен містити розмір масиву структур;
- pcEntriesRead – повинен містити покажчик на змінну, в яку запишеться кількість спільних ресурсів доступних на даний момент;
- pcTotalAvail – не використовується.

При розробці використовувались концепції діаграм діяльності. Тобто в UML, візуальне представлення графу діяльності. Граф діяльності є різновидом графу станів скінченного автомату, вершинами якого є певні дії, а переходи відбуваються по завершенню дій.

Це фундаментальна одиниця визначення поведінки в специфікації. Дія отримує множину вхідних сигналів, та перетворює їх на множину вихідних сигналів. Одна із цих множин, або обидві водночас, можуть бути порожніми. Виконання дії відповідає виконанню окремої дії. Подібно до цього, виконання діяльності є виконанням окремої діяльності, буквально, включно із виконанням тих дій, що містяться в діяльності. Кожна дія в діяльності може виконуватись один, два, або більше разів під час одного виконання діяльності. Щонайменше, дії мають отримувати дані, перетворювати їх та тестувати, деякі дії можуть вимагати певної послідовності. Специфікація діяльності (на вищих рівнях сумісності) може дозволяти виконання декількох (логічних) потоків, та існування механізмів синхронізації для гарантування виконання дій у правильному порядку.

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

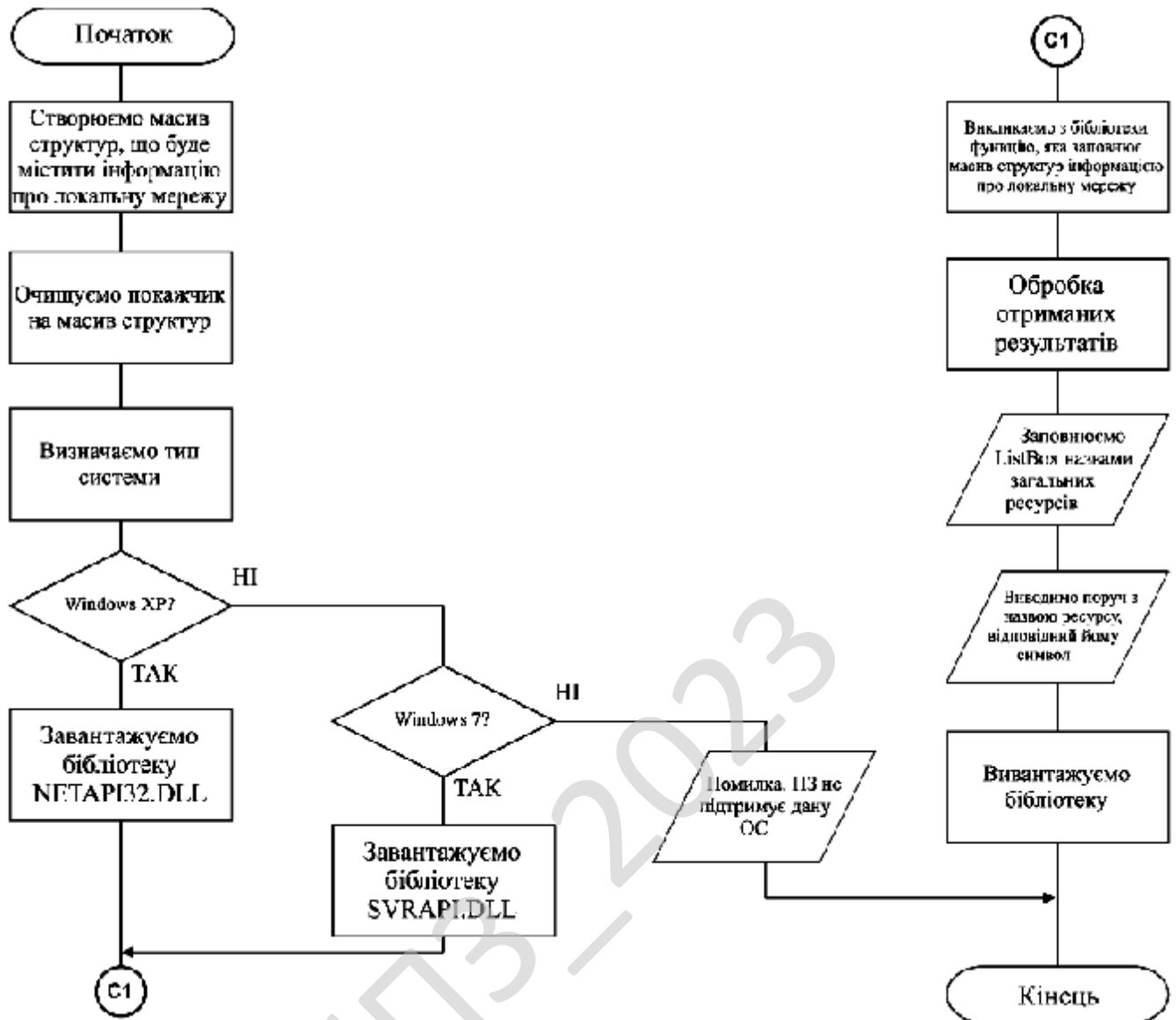


Рисунок 4.2 – Блок-схема підпрограми

Оголошення функції для Windows:

```

var
    NetShareEnum :function (ServerName :PWChar;
                            Level      :DWORD;
                            Bufptr    :Pointer;
                            Prefmaxlen :DWORD;
                            EntriesRead,
                            TotalEntries,
                            resume_handle:LPDWORD): DWORD; stdcall;
  
```



```

type
  TShareInfo50 = packed record
    shi50_netname : array [0..12] of Char;
    shi50_type    : Byte;
    shi50_flags   : Word;
    shi50_remark  : PChar;
    shi50_path    : PChar;
    shi50_rw_password : array [0..8] of Char;
    shi50_ro_password : array [0..8] of Char;
  end;

```

Поля:

- shi50_netname – містить рядок, що утримує мережне ім'я ресурсу;
- shi50_type – визначає тип ресурсу (докладніше в MSDN);
- shi50_flags – містить інформацію про права доступу до ресурсу;
- shi50_remark – покажчик на рядок, що містить необов'язковий коментар до ресурсу;
- shi50_path – містить локальне розташування ресурсу;
- shi50_rw_password – містить пароль на запис/читання;
- shi50_ro_password – містить пароль на читання.

Реально одержати значення двох останніх полів можна тільки при одержанні інформації про свій комп'ютер, в інших випадках вони залишаються порожніми.

Структура SHARE_INFO_2

Оголошення структури:

```

type
  TShareInfo2 = packed record
    shi2_netname      : PWChar;
    shi2_type         : DWORD;
    shi2_remark       : PWChar;
    shi2_permissions  : DWORD;
    shi2_max_uses     : DWORD;
    shi2_current_uses : DWORD;
    shi2_path         : PWChar;
    shi2_passwd       : PWChar;
  end;
  PShareInfo2 = ^TShareInfo2;

```

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

```
TShareInfo2Array = array [0..512] of TShareInfo2;  
PShareInfo2Array = ^TShareInfo2Array;
```

Поля:

- shi2_netname – містить покажчик на рядок, що утримує ім'я ресурсу;
- shi2_type – визначає тип ресурсу (докладніше в MSDN);
- shi2_remark – покажчик на рядок, що містить необов'язковий коментар до ресурсу;

- shi2_permissions – містить інформацію про права доступу до ресурсу;
- shi2_max_uses – визначає максимальну кількість підключень до ресурсу;
- shi2_current_uses – визначає кількість поточних підключень;
- shi2_path – містить покажчик на рядок утримуючий локальне розташування ресурсу;
- shi2_passwd – містить покажчик на рядок утримуючий пароль.

Закриття локального ресурсу

Тепер розглянемо функцію NetShareDel яка дозволяє закрити обраний загальний ресурс.

Оголошення функції для Windows 10/11:

```
var  
    NetShareDel: function (pszServer,  
                           pszNetName : PChar;  
                           usReserved : Word ): DWORD; stdcall;
```

Параметри:

- ServerName – повинен містити ім'я віддаленого комп'ютера, якщо закриваємо свої ресурси, то даному параметру потрібно привласнити NIL.
- NetName – покажчик на рядок, який містить ім'я ресурсу, що закривається.
- Reserved – не використовується, повинен бути рівним нулю.

Обидві функції не використовують ніяких структур. Нас цікавить тільки другий параметр, що містить ім'я ресурсу, що закривається. Як ім'я передається не шлях до ресурсу, а саме ім'я ресурсу яке ми визначили за допомогою коду даного вище. У випадку успішного виконання функцій, їхній результат буде рівний нулю.

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

Відкриття локального ресурсу

Під відкриттям локального ресурсу мається на увазі дозвіл загального доступу до нього. Блок-схема відкриття ресурсу зображена на рисунку 4.3.

Для вікриття локального ресурсу скористаємося функцією NetShareAdd.

Оголошення функції для Windows 10/11:

```
var
    NetShareAdd: function ( pszServer :Pchar;
                           SLevel      :Cardinal;
                           PbBuffer   :PChar;
                           CbBuffer   :Word):DWORD; stdcall;
```

Параметри:

– pszServer – повинен містити ім'я віддаленого комп'ютера, на якому повинна виконається функція, якщо відкриваємо локальний ресурс, то даному параметру потрібно привласнити NIL.

– sLevel – повинен містити ідентифікатор структури.

– pbBuffer – повинен містити покажчик на структуру.

– cbBuffer – повинен містити розмір структури.

Оголошення функції для Windows NT:

```
var
    NetShareAdd: function ( servername   : PWideChar;
                           level        : DWORD;
                           buf          : Pointer;
                           parm_err    : LPDWORD): DWORD; stdcall;
```

Параметри:

– servername – повинен містити ім'я віддаленого комп'ютера, на якому повинна виконається функція, якщо відкриваємо локальний ресурс то даному параметру потрібно привласнити NIL.

– level – повинен містити ідентифікатор структури.

– buf – повинен містити покажчик на структуру.

– parm_err – містить покажчик на член структури, що викликає помилку.

У функції для NT також використовується покажчик а не адреса покажчика. Обидві функції використовують структури описані вище.

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42


```

//Ім'я
    ShareNT.shi2_type := STYPE_DISKTREE;
//Тип ресурсу
    ShareNT.shi2_remark := ''; //Коментар
    ShareNT.shi2_permissions := ACCESS_READ;
//Доступ
    ShareNT.shi2_max_uses := DWORD(-1);
// Кількість макс. підключ.
    ShareNT.shi2_current_uses := 0;
// Кількість підкл.
    GetMem(TmpDirNT, TmpLength);
    StringToWideChar(TmpDir, TmpDirNT, TmpLength);
    ShareNT.shi2_path := TmpDirNT;
//Шлях до ресурсу
    ShareNT.shi2_passwd := nil;
//Пароль
    NetShareAddNT(nil, 2, @ShareNT, nil);
//Додаємо ресурс
    FreeMem (TmpNameNT);
//звільняємо пам'ять
    FreeMem (TmpDirNT);
end else begin //Код для 9x
    FLibHandle := LoadLibrary('SVRAPI.DLL');
    if FLibHandle = 0 then Exit;
    @NetShareAdd := GetProcAddress(FLibHandle, 'NetShareAdd');
    if not Assigned(NetShareAdd) then
    begin
        FreeLibrary(FLibHandle);
        Close;
    end;
    FillChar(Share9x.shi50_netname, SizeOf(Share9x.shi50_netname), #0);
    move(TmpName[1], Share9x.shi50_netname[0], Length(TmpName));
//Ім'я
    Share9x.shi50_type := STYPE_DISKTREE;
//Тип ресурсу
    Share9x.shi50_flags := SHI50F_RDONLY;
//Доступ
    FillChar(Share9x.shi50_remark,
        SizeOf(Share9x.shi50_remark), #0);
//Коментар
    FillChar(Share9x.shi50_path,
        SizeOf(Share9x.shi50_path), #0);

```

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

```

    Share9x.shi50_path := PAnsiChar(TmpDir);
//Шлях до ресурсу
    FillChar(Share9x.shi50_rw_password,
        SizeOf(Share9x.shi50_rw_password), #0);
//Пароль повного доступу
    FillChar(Share9x.shi50_ro_password,
        SizeOf(Share9x.shi50_ro_password), #0);
//Пароль для читання
    NetShareAdd(nil, 50, @Share9x, SizeOf(Share9x));
end;
FreeLibrary(FLibHandle);
end;

```

Змінні TmpNameNT і TmpDirNT звільняються тільки після виконання функції. Це критично, у протилежному випадку структура передана функції буде із двома "незаповненими" полями.

Також слід звернути увагу на те, що у версії коду для Windows 10/11 всі поля, що являють собою масив Char елементів, спочатку очищаються функцією FillChar, щоб уникнути перекручування даних (знак закінчення рядка дорівнює #0). Даний код відкриває новий ресурс на повний доступ.

Приховання й показ ресурсів

Сховати ресурс можна простим додаванням до його імені значка долара. Активувати, оберненою операцією. Це не зовсім правильно, але працює. Другий варіант сховати ресурс, це видалити його й створити його копію але із правами SHI50F_SYSTEM або вказівкою shi502_security_descriptor (для цього потрібно використовувати структуру SHARE_INFO_502).

Не можна проводити маніпуляції над наступними системними ресурсами: IPC\$, ADMIN\$, PRINT\$, WWWROOT\$, BIOSINFO\$, A\$, B\$, C\$, D\$, E\$, F\$, G\$, H\$, I\$, J\$, K\$, L\$, M\$, N\$, O\$, P\$, Q\$, R\$, S\$, T\$, U\$, V\$, W\$, X\$, Y\$, Z\$. Відкриття доступу до цих ресурсів робить беззахисною операційну систему.

Одержання списку поточних сесій

Для визначення користувачів підключених до нашого комп'ютера скористаємося функцією NetSessionEnum.

Оголошення функції для Windows 10/11:

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

```

var
  NetSessionEnum:function(   pszServer : PChar;
                             sLevel      : DWORD;
                             pbBuffer    : Pointer;
                             cbBuffer    : DWORD;
                             pcEntriesRead,
                             pcTotalAvial : Pointer):integer; stdcall;

```

Параметри:

- pszServer – повинен містити ім'я віддаленого комп'ютера, на якому повинна виконається функція, якщо дивимося в себе, то даному параметру потрібно привласнити NIL;
- sLevel – повинен містити ідентифікатор структури;
- pbBuffer – повинен містити покажчик на масив структур;
- cbBuffer – повинен містити розмір масиву структур;
- pcEntriesRead – покажчик на змінну утримуючу загальну кількість структур;
- pcTotalAvial – не використовується.

Оголошення функції для Windows NT:

```

var
  NetSessionEnum:function(   ServerName,
                             UncClientName,
                             Username   : PWChar;
                             Level       : DWORD;
                             bufptr      : Pointer;
                             pefmaxlen   : DWORD;
                             entriesread,
                             totalentries,
                             resume_handle : LPDWORD):DWORD; stdcall;

```

Параметри:

- ServerName – повинен містити ім'я віддаленого комп'ютера, на якому повинна виконається функція, якщо дивимося в себе, то даному параметру потрібно привласнити NIL.
- UncClientName – містить покажчик на рядок утримуючий ім'я сесії, про яку ми хочемо одержати інформацію, якщо потрібно переглянути всі сесії

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46


```

sesi50_idle_time   : Cardinal;
sesi50_protocol    : Byte;
pad1               : Byte;
end;

```

Поля:

– sesi50_cname – містить покажчик на рядок утримуючий ім'я комп'ютера, який встановив сесію;

– sesi50_username – містить покажчик на рядок утримуючий ім'я користувача, який встановив сесію;

– sesi50_key – містить значення за допомогою якого ми будемо завершувати сесію;

– sesi50_num_conns – містить число підключень зроблених під час сесії;

– sesi50_num_opens – містить кількість файлів відкритих під час сесії;

– sesi50_time – містить час в секундах, протягом якого сесія була активна;

– sesi50_idle_time – містить час у секундах протягом якого сесія була неактивна;

– sesi50_protocol – містить ім'я протоколу, за допомогою якого клієнт зв'язується із сервером;

– Pad1 – невеликий вирівнювач структури, не використовується.

Структура SESSION_INFO_502

Опис структури:

```

type
  TSessionInfo502 = packed record
    Sesi502_cname      : PWideChar;
    Sesi502_username  : PWideChar;
    Sesi502_num_opens : DWORD;
    Sesi502_time      : DWORD;
    Sesi502_idle_time : DWORD;
    Sesi502_user_flags: DWORD;
    Sesi502_cltype_name : PWideChar;
    Sesi502_transport : PWideChar;
  End;
  PSessionInfo502 = ^TSessionInfo502;
  TSessionInfo502Array = array[0..512] of TSessionInfo502;

```

```
PSessionInfo502Array = ^TSessionInfo502Array;
```

Поля:

– sesi502_cname – містить покажчик на рядок утримуючий ім'я комп'ютера, який встановив сесію;

– sesi502_username – містить покажчик на рядок утримуючий ім'я користувача, який встановив сесію;

– sesi502_num_opens – містить кількість файлів відкритих під час сесії;

– sesi502_time – містить час у секундах протягом якого сесія була активна;

– sesi502_idle_time – містить час у секундах протягом якого сесія була неактивна;

– sesi50_user_flags – значення, що описує як користувач установив сесію;

– sesi502_cltype_name – тип клієнта, який встановив сесію, не використовується;

– sesi502_transport – містить ім'я протоколу, за допомогою якого клієнт зв'язується із сервером.

Розглянемо детальніше значення полів time і idle_time. Що означає коли сесія не активна? Наприклад Ви відкрили якийсь ресурс на віддаленій машині, просто подивитися, що в ньому знаходиться (наприклад імена файлів). У той час коли Ви нічого не робите, не копіюєте, не запускаєте, не відкриваєте файли, сесія не активна, вона чекає ваших дій. Як тільки Ви починаєте копіювати файл з віддаленої машини, сесія стає активною до закінчення копіювання.

Слід звернути увагу на поле key структури TSessionInfo50, воно містить унікальний ідентифікатор за допомогою якого можна завершити сесію.

Код одержання поточних сесій:

```
procedure TMainForm.btnGetSessionsClick(Sender: TObject);
var
  OS: Boolean;
  FLibHandle : THandle;
  SessionInfo50: array [0..512] of TSessionInfo50;
  SessionInfo502 : PSessionInfo502Array;
  TotalEntries, EntriesReadNT: DWORD;
  EntriesRead, TotalAvial: Word;
```

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

```

    i:integer;
begin
    lvSessions.Items.Clear;
    if not IsNT(OS) then Close;
//З'ясовуємо тип системи
    if OS then begin //Код для NT
        FLibHandle := LoadLibrary('NETAPI32.DLL');
        if FLibHandle = 0 then Exit;
        @NetSessionEnumNT := GetProcAddress(FLibHandle, 'NetSessionEnum');
        if not Assigned(NetSessionEnumNT) then
            begin
                FreeLibrary(FLibHandle);
                Exit;
            end;
        SessionInfo502 := nil;
        if
            NetSessionEnumNT(nil,nil,nil,502,@SessionInfo502,DWORD(-
1),@entriesreadNT, @totalentries, nil)=0 then
            for i:=0 to EntriesReadNT-1 do
                begin
                    with lvSessions.Items.Add do
//Заповнення даними зі структури
                        begin
                            Caption := string(SessionInfo502^[i].sesi502_cname);
//Ім'я комп'ютера
                            SubItems.Add(SessionInfo502^[i].sesi502_username);
//Ім'я користувача
                            SubItems.Add(IntToStr(SessionInfo502^[i].sesi502_num_opens));
//Відкритих ресурсів
                            SubItems.Add(CardinalToTimeStr(SessionInfo502^[i].Sesi502_Time));
//Час активн.
SubItems.Add(CardinalToTimeStr(SessionInfo502^[i].sesi502_idle_time));
//Час не активний
                        end;
                    end;
                end else begin //Код для Windows 10/11
                    FLibHandle := LoadLibrary('SVRAPI.DLL');
                    if FLibHandle <> 0 then Exit;
                    @NetSessionEnum := GetProcAddress(FLibHandle, 'NetSessionEnum');
                    if not Assigned(NetSessionEnum) then
                        begin
                            FreeLibrary(FLibHandle);
                            Exit;
                        end;
                    end;
                end
            end
        end
    end
end

```

						ВКРМ-122.23.0079.00.00.ПЗ	<i>Арк.</i>
<i>Вим.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>			50

```

end;
if NetSessionEnum (nil,50,@SessionInfo50,SizeOf(SessionInfo50),
@EntriesRead,@TotalAvial) = 0 then
for i:=0 to EntriesRead-1 do
begin
with lvSessions.Items.Add do //Заповнення даними зі структури
begin
Caption := string(SessionInfo50[i].Sesi50_cname);
//Ім'я комп'ютера
SubItems.Add(SessionInfo50[i].Sesi50_username);
//Ім'я користувача
SubItems.Add(IntToStr(SessionInfo50[i].sesi50_num_opens));
//Відкритих ресурсів
SubItems.Add(CardinalToTimeStr(SessionInfo50[i].Sesi50_Time));
//Час активн.
SubItems.Add(CardinalToTimeStr(SessionInfo50[i].sesi50_idle_time));
//Час не активн.
SessionCloseKey[i]:= SessionInfo50[i].sesi50_key;
//Унікальний ідентифікатор для закриття
end;
end;
end;
FreeLibrary(FLibHandle);
end;

```

Ключі для закриття сесій слід заносити в масив у тому порядку, в якому були отримані самі сесії. Це зроблено для простоти. Якщо в списку, що відображає сесії, не застосовувати сортування, то порядковий номер виділеної для закриття сесії і її ідентифікатор у масиві збіжаться.

Завершення сесій

Для завершення відкритих сесій будемо використовувати функцію NetSessionDel.

Оголошення функції для Windows 10/11:

```

var
    NetSessionDel:function( pszServer : PChar;
                           PszClientName : PChar;
                           SReserved : SmallInt):DWORD; stdcall;

```

Параметри:

– pszServer – повинен містити ім'я віддаленого комп'ютера, на якому

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

повинна виконається функція, якщо завершується сесія в себе, то даному параметру потрібно привласнити NIL;

- pszClientName – повинен містити ім'я клієнта, чия сесія завершується;
- sReserved – повинен містити унікальний ключ для завершення сесії (той який ми одержали попередньою функцією).

Оголошення функції для Windows NT:

```
var  
    NetSessionDel:function( ServerName,  
                            UncClientName,  
                            Username :PWChar):DWORD; stdcall;
```

Параметри:

- ServerName – повинен містити ім'я віддаленого комп'ютера, на якому повинна виконається функція, якщо завершується сесія в себе, то даному параметру потрібно привласнити NIL;
- uncClientName – повинен містити ім'я клієнта, чия сесія завершується, якщо параметр NIL, завершаться всі сесії зазначені в параметрі username;
- username – повинен містити ім'я користувача, чия сесія завершується, якщо параметр NIL, завершаться всі сесії зазначені в параметрі uncClientName.

Як можна помітити, ніяких структур дані функції не використовують. Напишемо процедуру яка буде завершувати обрану нами в lvSessions сесію по ім'ю клієнта. От як цей код виглядає:

```
procedure TMainForm.btnCloseSessionClick(Sender: TObject);  
var  
    OS: Boolean;  
    FLibHandle : THandle;  
    CNameNT: PWideChar;  
    CName9x: PAnsiChar;  
    Key:SmallInt;  
    i: Integer;  
begin  
    if not IsNT(OS) then Close;  
    //З'ясовуємо тип системи  
    if not Assigned(lvSessions.Selected) then Exit;  
    i:= lvSessions.Selected.Index; //Визначаємо номер обраної сесії
```

```

if OS then begin
  FLibHandle := LoadLibrary('NETAPI32.DLL');
  if FLibHandle = 0 then Exit;
  @NetSessionDelNT := GetProcAddress(FLibHandle, 'NetSessionDel');
  if not Assigned(NetSessionDelNT) then
  begin
    FreeLibrary(FLibHandle);
    Exit;
  end;
  //Перетворимо дані в необхідний вид
  CNameNT := PWChar(WideString('\'+lvSessions.Items.Item[i].Caption));
  NetSessionDelNT(nil,CNameNT,nil);
end else begin
  FLibHandle := LoadLibrary('SVRAPI.DLL');
  if FLibHandle = 0 then Exit;
  @NetSessionDel := GetProcAddress(FLibHandle, 'NetSessionDel');
  if not Assigned(NetSessionDel) then
  begin
    FreeLibrary(FLibHandle);
    Exit;
  end;
  //Перетворимо дані в необхідний вид
  CName9x := PAnsiChar(lvSessions.Items.Item[i].Caption);
  key := SessionCloseKey[i]; //Беремо ключ із масиву
  NetSessionDel(nil,CName9x,Key);
end;
FreeLibrary(FLibHandle);
end;

```

Одержання списку відкритих файлів

Розглянемо функцію NetFileEnum.

Оголошення функції для Windows 10/11:

```

var
  NetFileEnum:function( pszServer,
                        pszBasePath : PChar;
                        sLevel       : DWORD;
                        pbBuffer    : Pointer;
                        cbBuffer    : DWORD;
                        pcEntriesRead,
                        pcTotalAvail : Pointer):Integer; stdcall;

```

Параметри:

- pszServer – повинен містити ім'я віддаленого комп'ютера, на якому повинна виконається функція, якщо дивимося в себе, то даному параметру потрібно привласнити NIL;
- pszBasePath – повинен містити каталог, відкриті файли з якого ми хочемо подивитися, якщо ми хочемо переглянути всі відкриті файли параметру потрібно привласнити NIL;
- sLevel – повинен містити ідентифікатор структури;
- pbBuffer – повинен містити покажчик на масив структур;
- cbBuffer – повинен містити розмір структури;
- pcEntriesRead – покажчик на змінну утримуючу загальну кількість структур;
- pcTotalAvial – не використовується.

Оголошення функції для Windows NT:

```
var
    NetFileEnum:function(  servername,
                           basepath,
                           username  : PWChar;
                           level      : DWORD;
                           bufptr    : Pointer;
                           pefmaxlen  : DWORD;
                           entriesread,
                           totalentries,
                           resume_handle : LPDWORD):DWORD; stdcall;
```

Параметри:

- ServerName – повинен містити ім'я віддаленого комп'ютера, на якому повинна виконається функція, якщо дивимося в себе, то даному параметру потрібно привласнити NIL.
- BasePath – повинен містити каталог, відкриті файли з якого ми хочемо подивитися, якщо ми хочемо переглянути всі відкриті файли параметру потрібно привласнити NIL.
- UserName – містить покажчик на рядок утримуючий ім'я користувача,


```

begin
  lvfiles.Items.Clear;
  if not IsNT(OS) then Close;
  //З'ясовуємо тип системи
  if OS then begin //Код для NT
    FLibHandle := LoadLibrary('NETAPI32.DLL');
    if FLibHandle = 0 then Exit;
    @NetFileEnumNT := GetProcAddress(FLibHandle, 'NetFileEnum');
    if not Assigned(NetFileEnumNT) then
      begin
        FreeLibrary(FLibHandle);
        Exit;
      end;
    FileInfoNT := nil;
    if NetFileEnumNT(nil, nil, nil, 3, @FileInfoNT, DWORD(-1), @entriesreadNT,
@totalentries, nil)=0 then
      for i:=0 to EntriesReadNT-1 do
        begin
          with lvFiles.Items.Add do
            //Заповнення даними зі структури
            begin
              Caption := string(IntToStr(FileInfoNT^[i].fi3_id));
            //Ідентифікатор
              SubItems.Add(FileInfoNT^[i].fi3_pathname);
            //Шлях до файлу
              SubItems.Add(FileInfoNT^[i].fi3_username);
            //Ім'я користувача
            end;
          end;
        end else begin
          FLibHandle := LoadLibrary('SVRAPI.DLL');
          if FLibHandle = 0 then Exit;
          @NetFileEnum := GetProcAddress(FLibHandle, 'NetFileEnum');
          if not Assigned(NetFileEnum) then
            begin
              FreeLibrary(FLibHandle);
              Exit;
            end;
          if NetFileEnum (nil,
nil, 50, @FileInfo9x, SizeOf(FileInfo9x), @EntriesRead, @TotalAvial) = 0 then
            for i:=0 to EntriesRead-1 do
              begin

```

```

        with lvFiles.Items.Add do
//Заповнення даними зі структури
        begin
            Caption := string(IntToStr(FileInfo9x[i].fi50_id));
//Ідентифікатор
            SubItems.Add(FileInfo9x[i].fi50_pathname);
//Шлях до файлу
            SubItems.Add(FileInfo9x[i].fi50_username);
//Ім'я користувача
        end;
    end;
end;
FreeLibrary(FLibHandle);
end;

```

Закриття відкритого файлу

Тепер розглянемо функції NetFileClose і NetFileClose2, за допомогою яких будемо закривати відкриті по мережі файли.

Функція NetFileClose2 – використовується тільки в Windows 10/11:

```

var
NetFileClose2:function(pszServer :PChar;
                        UlFileId :LongWord):DWORD; stdcall;

```

Поля:

- pszServer – повинен містити ім'я віддаленого комп'ютера, якщо закриваємо в себе, то даному параметру потрібно привласнити NIL.
- UlFileId – повинен містити унікальний ідентифікатор відкритого файлу.

Функція NetFileClose – використовується тільки в Windows NT:

```

var
NetFileClose:function( ServerName :PWideChar;
                        FileId :DWORD):DWORD; stdcall;

```

Поля:

- ServerName – повинен містити ім'я віддаленого комп'ютера, якщо закриваємо в себе, то даному параметру потрібно привласнити NIL.
- FileId – повинен містити унікальний ідентифікатор відкритого файлу.

Як перший параметр передамо NIL (закриваємо в себе), у якості другого, ідентифікатор файлу, що відображається у полі ID. От приклад коду:

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

```

procedure TMainForm.btnCloseFileClick(Sender: TObject);
var
  OS: Boolean;
  FLibHandle : THandle;
  i: Integer;
begin
  if not IsNT(OS) then Close;
  //З'ясуємо тип системи
  if not Assigned(lvFiles.Selected) then Exit;
  i:= lvFiles.Selected.Index;
  //Визначаємо номер обраного файлу
  if OS then begin //Код для NT
    FLibHandle := LoadLibrary('NETAPI32.DLL');
    if FLibHandle = 0 then Exit;
    @NetFileClose := GetProcAddress(FLibHandle, 'NetFileClose');
    if not Assigned(NetFileClose) then
      begin
        FreeLibrary(FLibHandle);
        Exit;
      end;
    NetFileClose(nil, StrToInt(lvFiles.Items.Item[i].Caption));
  //Закриваємо
  end else begin //Код для Windows 10/11
    FLibHandle := LoadLibrary('SVRAPI.DLL');
    if FLibHandle = 0 then Exit;
    @NetFileClose2 := GetProcAddress(FLibHandle, 'NetFileClose2');
    if not Assigned(NetFileClose2) then
      begin
        FreeLibrary(FLibHandle);
        Exit;
      end;
    NetFileClose2(nil, StrToInt(lvFiles.Items.Item[i].Caption));
  //Закриваємо
  end;
  FreeLibrary(FLibHandle);
end;

```

Визначення вхідного та вихідного трафіку

Для цього досить використовувати всього лише одну функцію бібліотеки IPHLPAPI.DLL, що поставляється з усіма версіями Windows. Оголошення функції (всі версії Windows):

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

```

var
  GetIfTable:function(  pIfTable: PMibIfTable;
                        pdwSize  : PULONG;
                        bOrder   : Boolean ): DWORD; stdcall;

```

Параметри:

- pIfTable – повинен містити покажчик на структуру.
- pdwSize – повинен містити розмір структури.
- bOrder – указує, чи потрібне сортування в масиві, що повертається.

Як перший параметр функція використовує покажчик на структуру. Опис структури:

```

type
  TMibIfTable = packed record
    dwNumEntries : DWORD;
    Table        : TMibIfArray;
  end;
  PMibIfTable = ^ TMibIfTable;

```

Поля:

- dwNumEntries – визначає розмірність масиву представленого другим параметром.
- Table – є масивом структур.

Структура сама по собі вкрай неінформативна, нас цікавить друге її поле, що також представляє собою структуру. Опис структури:

```

type
  TMibIfRow = packed record
    wszName      : array[0..255] of WideChar;
    dwIndex      : DWORD;
    dwType       : DWORD;
    dwMtu        : DWORD;
    dwSpeed      : DWORD;
    dwPhysAddrLen : DWORD;
    bPhysAddr    : array[0..7] of Byte;
    dwAdminStatus : DWORD;
    dwOperStatus : DWORD;
    dwLastChange : DWORD;
    dwInOctets   : DWORD;
    dwInUcastPkts : DWORD;
    dwInNUCastPkts : DWORD;

```

```

dwInDiscards : DWORD;
dwInErrors      : DWORD;
dwInUnknownProtos : DWORD;
dwOutOctets     : DWORD;
dwOutUCastPkts  : DWORD;
dwOutNUCastPkts : DWORD;
dwOutDiscards   : DWORD;
dwOutErrors     : DWORD;
dwOutQLen       : DWORD;
dwDescrLen      : DWORD;
bDescr          : array[0..255] of Char;
end;
TMibIfArray = array [0..512] of TMibIfRow;
PMibIfRow = ^TMibIfRow;
PmibIfArray = ^TmibIfArray;

```

Поля:

- wszName – покажчик на рядок утримуючий ім'я інтерфейсу;
- dwIndex – визначає індекс інтерфейсу;
- dwType – визначає тип інтерфейсу;
- dwMtu – визначає максимальну швидкість передачі;
- dwSpeed – визначає поточну швидкість передачі в бітах у секунду;
- dwPhysAddrLen – визначає довжину адреси, що втримується в bPhysAddr;
- bPhysAddr – містить фізичну адресу інтерфейсу (його трохи видозмінений MAC-адрес);
- dwAdminStatus – Визначає активність інтерфейсу;
- dwOperStatus – містить поточний статус інтерфейсу;
- dwLastChange – містить останній змінений статус;
- dwInOctets – містить кількість байтів прийнятих через інтерфейс;
- dwInUcastPkts – містить кількість спрямованих пакетів прийнятих інтерфейсом;
- dwInNUCastPkts – містить кількість ненаправлених пакетів прийнятих інтерфейсом;

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

- dwInDiscards – містить кількість забракованих вхідних пакетів (навіть якщо вони не містили помилки);
- dwInErrors – містить кількість вхідних пакетів утримуючих помилки;
- dwInUnknownProtos – містить кількість забракованих вхідних пакетів зі структурою невідомого протоколу;
- dwOutOctets – містить кількість байтів відправлених інтерфейсом;
- dwOutUCastPkts – містить кількість спрямованих пакетів відправлених інтерфейсом;
- dwOutNUCastPkts – містить кількість ненаправлених пакетів відправлених інтерфейсом;
- dwOutDiscards- містить кількість забракованих вихідних пакетів (навіть якщо вони не містили помилки);
- dwOutErrors – містить кількість вихідних пакетів утримуючих помилки;
- dwOutQLen – містить довжину черги даних;
- dwDescrLen – містить розмір масиву bDescr;
- bDescr – містить опис інтерфейсу.

В цій структурі утримується безліч інформації, що ми й будемо використовувати.

Код визначення поточного вхідного/вихідного трафіку виглядає наступним чином:

```

procedure TMainForm.tmrTrafficTimer(Sender: TObject);
// Допоміжна функція, що перетворить MAC-адресу до "нормального" виду
//Визначаємо спеціальний тип, щоб можна було передати у функцію масив
type TMAC = array [0..7] of Byte;
//Як перше значення масив, друге значення, розмір даних у масиві
function GetMAC(Value: TMAC; Length: DWORD): String;
var
  i: Integer;
begin
  if Length = 0 then Result := ' 00-00-00' else
  begin
    Result := '';
    for i:= 0 to Length -2 do
  
```

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

```

        Result := Result + IntToHex(Value[i],2)+'-';
        Result := Result + IntToHex(Value[ Length-1],2);
    end;
end;

//Сама процедура
var
    FLibHandle      : THandle;
    Table          : TMibIfTable;
    i              : Integer;
    Size           : Integer;
begin
    tmrTraffic.Enabled := False;
//Зупиняємо таймер
    lvTraffic.Items.BeginUpdate;
    lvTraffic.Items.Clear;
//Очищаємо список
    FLibHandle := LoadLibrary('IPHLPAPI.DLL');
//Завантажуємо бібліотеку
    if FLibHandle = 0 then Exit;
    @GetIfTable := GetProcAddress(FLibHandle, 'GetIfTable');
    if not Assigned(GetIfTable) then
    begin
        FreeLibrary(FLibHandle);
        Close;
    end;
    Size := SizeOf(Table);
    if GetIfTable(@Table, @Size, False ) = 0 then
//Виконуємо функцію
        for i:= 0 to Table.dwNumEntries-1 do begin
            with lvTraffic.Items.Add do begin
//Виводимо результати
                Caption := String(Table.Table[i].bDescr);
//Найменування інтерфейсу
                SubItems.Add(GetMAC (TMAC (Table.Table[i].bPhysAddr) ,
                    Table.Table[i].dwPhysAddrLen)); //MAC адреса
                SubItems.Add(IntToStr (Table.Table[i].dwInOctets));
//Усього прийнято байтів
                SubItems.Add(IntToStr (Table.Table[i].dwOutOctets));
//Усього відправлено байтів
            end;
        end;
    end;
end;

```

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

```

lvTraffic.Items.EndUpdate;
FreeLibrary(FLibHandle);
tmrTraffic.Enabled := True;
//Активуємо таймер
end;

```

Тип даних TMAC було створено для передачі масиву, у якому втримується сам MAC адресу у функцію для перетворення його в більш звичний вигляд.

Код TMAC(Table.Table[i].bPhysAddr) – це передача масиву, обов'язково потрібно вказати, що масив передається як тип TMAC, у протилежному випадку компілятор видасть помилку несумісності типів.

4.2 Захист розробленого програмного забезпечення

Захист розробленого програмного забезпечення буде відбуватися за допомогою алгоритму DES.

DES є класичною мережею Фейштеля із двома гілками. Дані шифруються 64-бітними блоками, використовуючи 56-бітний ключ. Алгоритм перетворить за кілька раундів 64-бітний вхід в 64-бітний вихід. Довжина ключа дорівнює 56 бітам. Процес шифрування складається із чотирьох етапів. На першому з них виконується початкова перестановка (IP) 64-бітного вихідного тексту (забілювання), під час якої біти переставляються у відповідності зі стандартною таблицею. Наступний етап складається з 16 раундів однієї й тої ж функції, що використовує операції зрушення й підстановки. На третьому етапі ліва й права половини виходу останньої (16-й) ітерації міняються місцями. Нарешті, на четвертому етапі виконується перестановка IP^{-1} результату, отриманого на третьому етапі. Перестановка IP^{-1} інверсна початковій перестановці.

Праворуч на рисунку показаний спосіб, яким використовується 56-бітний ключ. Спочатку ключ подається на вхід функції перестановки. Потім для кожного з 16 раундів підключ K_i є комбінацією лівого циклічного зрушення й перестановки. Функція перестановки та сама для кожного раунду, але підключи

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

K_i для кожного раунду виходять різні внаслідок повторюваного зрушення біт ключа.



Рисунок 4.3 – Загальна схема DES

КБПЗ_2023

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розглянемо розроблене ПЗ яке зображено на рисунку 5.1. Як видно з рисунку, користувач може переглянути ресурси мережі, відкриті сесії, відкриті користувачами мережі файли, мережні інтерфейси, вхідний та вихідний трафік.

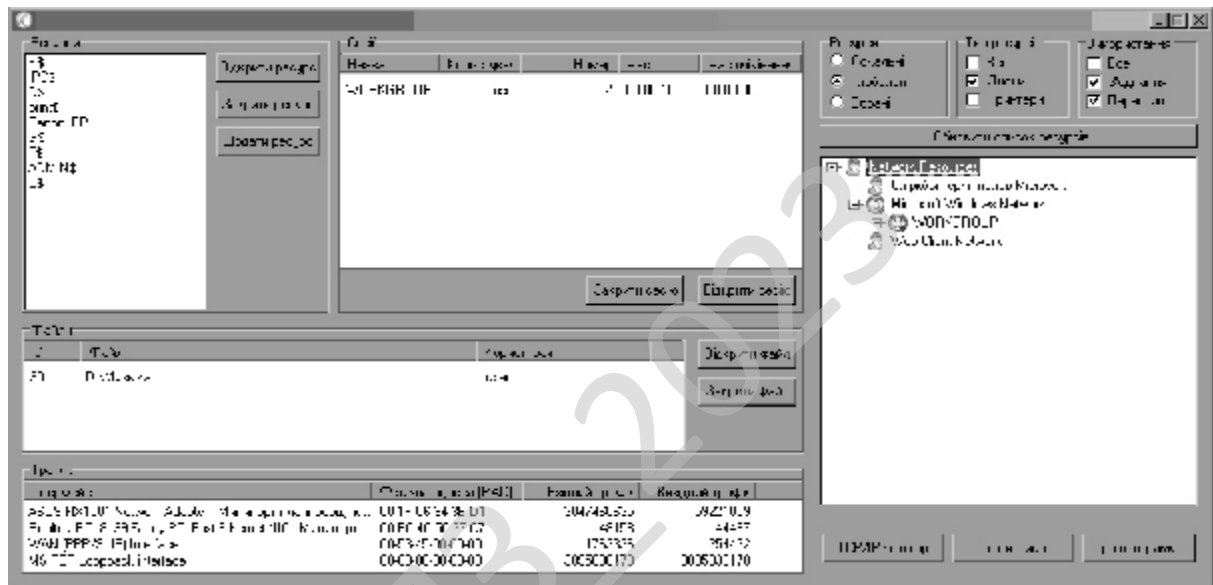


Рисунок 5.1 – Головне вікно ПЗ

Також користувач розробленої програми може виконувати наступні дії:

- Відкрити ресурс.
- Закрити ресурс.
- Додати ресурс.
- Обновити список ресурсів.
- Закрити сесію.
- Відкрити сесію.
- Відкрити файл.
- Закрити файл.

- Ввімкнути TSP/IP монітор.
- Переглянути статистику з'єднань.
- Переглянути довідку.

На рисунку 5.2 показане вікно монітору TSP/IP з'єднань. Для його виклику слід натиснути кнопку " TSP/IP монітор".

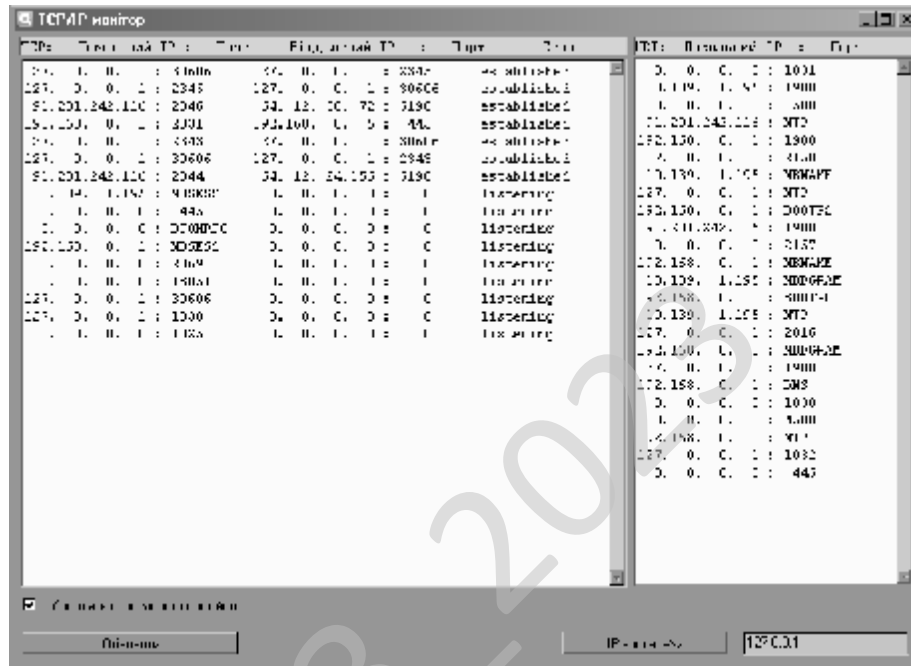


Рисунок 5.2 – Монітор TSP/IP з'єднань

Монітор TSP/IP з'єднань дозволяє переглядати всі з'єднання, що відбуваються у мережі. Монітор складається з двох полів, в першому показані з'єднання по TSP-протоколу, в другому – по UDP-протоколу. Також при необхідності можна перешлянути вікно статистики.

Для його виклику слід натиснути кнопку "Статистика". Вікно статистики складається з п'яти полів: Поле статистики подій по TSP-протоколу; Поле статистики подій по IP-протоколу; Поле статистики подій по UDP-протоколу; Поле статистики вхідних подій по ICMP-протоколу; Поле статистики вихідних подій по ICMP-протоколу.

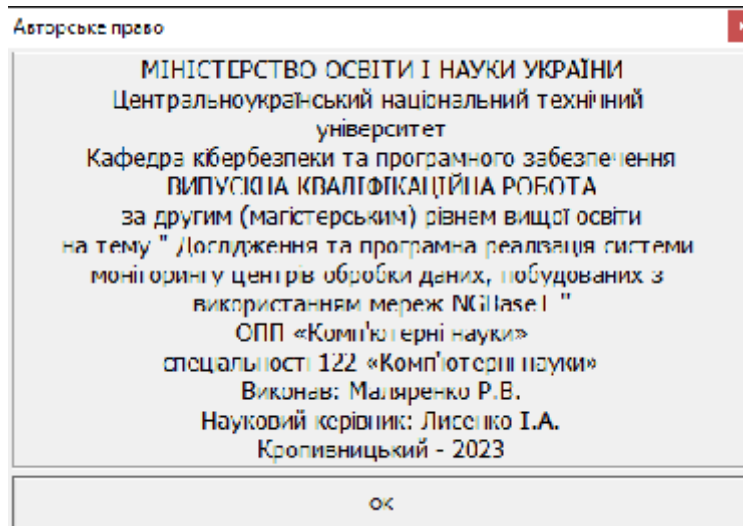


Рисунок 5.3 – Авторське право

На рисунку 5.3 зображено авторські дані розробленого програмного забезпечення. Обрано умови розповсюдження – Freeware.

Це власницьке програмне забезпечення, котре можна безоплатно використовувати протягом необмеженого терміну без обмежень у функціональності, і поширюване без сирцевих кодів.

Автори такого програмного забезпечення, як правило, хочуть «дати щось спільноті», але хочуть також контролювати його подальшу розробку. Іноді, коли програмісти вирішують припинити розробку, вони передають сирцевий код іншим програмістам, або ж спільноті як вільне програмне забезпечення. Дуже часто плутають поняття «безоплатне програмне забезпечення» та «вільне програмне забезпечення», хоча вони суттєво відрізняються. Безоплатне програмне забезпечення можна безоплатно встановлювати та використовувати (іноді з певними обмеженнями, як, наприклад, «безоплатне для домашнього або некомерційного вжитку»), в той час як вільне програмне забезпечення можна продавати за будь-яку суму, але при тому, у користувача, котрий його отримує, повинні бути права на вивчення, модифікацію та поширення сирцевих кодів одержаної програми.

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи моніторингу центрів обробки даних, побудованих з використанням мереж NGBaseT.

Метою розробки є дослідження та програмна реалізація системи моніторингу центрів обробки даних, побудованих з використанням мереж NGBaseT.

Об'єктом дослідження є процес моніторингу центрів обробки даних, побудованих з використанням мереж NGBaseT.

Предметом дослідження є методи моніторингу центрів обробки даних, побудованих з використанням мереж NGBaseT.

Методи дослідження базуються на методах хмарних технологій, методах математичної статистики, методах розробки програмного забезпечення.

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

- Удосконалено метод моніторингу центрів обробки даних, побудованих з використанням мереж NGBaseT.
- Розроблено вітчизняний продукт моніторингу центрів обробки даних, побудованих з використанням мереж NGBaseT, який має більш широкі можливості, на відміну від існуючих аналогів.

					VKPM-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

7 ДАНІ ПРО ЕКОНОМІЧНУ ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ

7.1 Техніко-економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти

Після ознайомлення з підприємством та засобами розробки програмної продукції був розроблений план розробки програми. Був підрахований необхідний час для розробки та впровадження програми. Цей час склав 48 днів (два місяці).

В магістерській роботі було проведено дослідження та виконана програмна реалізація системи моніторингу центрів обробки даних, побудованих з використанням мереж NGBaseT. Розроблене програмне забезпечення має достатню надійність і задовольняє усім поставленим умовам, а саме:

- а) невеликий розмір;
- б) невеликі системні потреби;
- в) незалежність від встановлених на комп'ютері баз даних;
- г) зручність у користуванні та надійність.

Таблиця 7.1 – Початкові дані

Показники	Позначення	Характеристика або величина
1	2	3
1. Кількість розроблених програм період, шт.	N	1
2. Кількість екземплярів програм, шт.	Ne	17
3. Запланований термін розробки, днів	Fpq	48 (2 місяці)
4. Група задачі підсистеми управління (1-6)	–	1
5. Ступінь новизни задачі (А, Б, В, Г)	–	Б
6. Складність алгоритму (1, 2, 3)	–	2

Продовження таблиці 7.1

1	2	3
7. Кількість макетів вхідної інформації	–	7
8. Кількість форм вихідної інформації.	–	8
9. Мова програмування (1-6)	–	2
10. Попередній досвід (1-6)	–	2
11. Гнучкість проекту ПП (1-6)	–	3
12. Детальність проекту ПП (1-6)	–	2
13. Рівень спрацьованості колективу (1-6)	–	1
14. Ступінь вимірності процесів (1-6)	–	3
15. Необхідна надійність програмного забезпечення (1-6)	–	2
16. Розмір бази даних (порівняно з розміром програми) (1-6)	–	3
17. Складність кінцевого програмного продукту (1-6)	–	2
18. Необхідний рівень забезпечення повторного використання (1-6)	–	1
19. Документованість відповідно до планованого життєвого циклу (1-6)	–	1
20. Вимоги до швидкодії ПП (1-6)	–	2
21. Обмеження на розміри основного сховища даних (1-6)	–	2
22. Різноманітність використовуваних обчислювальних платформ (1-6)	–	1
23. Професійний рівень аналітиків (1-6)	–	2
24. Професійний рівень програмістів (1-6)	–	3
25. Постійність складу команди розробників (1-6)	–	2
26. Досвід розробки додатків (1-6)	–	3
27. Досвід роботи з обчислювальною платформою (1-6)	–	2

Продовження таблиці 7.1

1	2	3
28. Досвід роботи з мовою і інструментами середовища розробки (1-6)	–	1
29. Досвід роботи з програмними інструментами розробки (1-6)	–	2
30. Розробка ПЗ для декількох серверів одночасно (1-6)	–	3
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн.	–	17000
33. Норматив додаткової зарплати, % :	Н _д	10
34. Норматив відрахувань у соціальні фонди, %	Н _с	22
35. Норматив загальногосподарських витрат, %	Н _г	15
36. Норматив витрат на освоєння нових мов програмування, %	Н _п	15
37. Рівень рентабельності програмної продукції, %	Р _е	50
38. Ставка податку на додану вартість, %	Н _{дв}	20

7.2 Розрахунок трудомісткості розробки програмної продукції

Значення трудомісткості розробки програмного забезпечення для стадій ТЗ, ЕК, ТП та ВП визначаємо по типовим нормам часу приведеним в додатках МВ. Стадія РП є найбільш тривалою і трудомісткою, що робить значний вплив на інші стадії проекту.

Визначимо трудомісткість розробки ПЗ для стадії РП.

Обчислюємо номінальні трудовитрати, люд-міс.:

$$T_{ном} = A \text{ Size}^B, \quad (7.1)$$

де: A – коефіцієнт Боема, $A = 2,45$;

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

Size – загальний об'єм відлагодженого програмного коду, тис. рядків;

B – показник ступеня, що визначається співвідношенням:

$$B = 1,01 + 0,001 \sum W_i, \quad (7.2)$$

де: W_i – сумарне значення п'яти показників (МВ, додаток 2), що відображають особливості розробки проекту програмного продукту (ПП) і колективу розробників.

$$B = 1,01 + 0,001(3,24 + 3,64 + 3,38 + 4,94 + 2,73) = 1,028.$$

$$T_{ном} = 2,45 \cdot 2,7^{1,028} = 6,8 \text{ люд-міс.}$$

Визначаємо уточнені (з урахуванням приведених в МВ додатку 3 сімнадцяти додаткових коефіцієнтів) трудовитрати, люд-міс.:

$$T_{уточн} = T_{ном} PV_j, \quad (7.3)$$

де: PV_j – добуток сімнадцяти додаткових коефіцієнтів, приведених в МВ додатку 3.

$$T_{уточн} = 6,8 \cdot (0,88 \cdot 1 \cdot 0,88 \cdot 0,91 \cdot 0,89 \cdot 1 \cdot 1 \cdot 0,87 \cdot 1,22 \cdot 1 \cdot 1,1 \cdot 1 \cdot 1,12 \cdot 1,22 \cdot 1,12 \cdot 1 \cdot 1,1) = 8,38 \text{ люд-міс.}$$

Ці коефіцієнти дозволяють диференційовано оцінювати результати роботи програмістів, беручи до уваги швидкодію програми, використання різноманітних обчислювальних платформ і інструментів розробки, взаємодію декількох серверів, вимоги до об'ємів баз даних і ін.

Визначаємо підсумкові трудовитрати по стадії робочий проект, люд-дні:

$$T_{РП} = 0,3 C T_{уточн}^{0,33 + 0,2(B-1,01)} S, \quad (7.4)$$

де: C – визначений емпірично коефіцієнт, запропонований авторами методики, (МВ, додаток 4). Для мови програмування Delphi – 2,66;

S – коефіцієнт стиснення (або подовження) графіка робіт %, що дозволяє коректувати терміни розробки ПЗ згідно встановленим вимогам. Вибираємо в межах (25...350)%.

$$T_{РП} = 0,3 \cdot 2,66 \cdot 8,38^{0,33 + 0,2(1,028 - 1,01)} \cdot 100 = 197 \text{ люд/день.}$$

Для зручності визначення загальної трудомісткості на розробку програмного забезпечення результати розрахунків по стадіям зводимо до таблиці 7.2.

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

Таблиця 7.2 – Визначення трудомісткості розробки програмного забезпечення

Стадії розробки	Трудомісткість за типовими нормами та розрахунками	
	Величина, люд/дні	Підстава
Технічне завдання	9	Д5
Ескізний проект	10	Д6
Технічний проект	15	Д7
Робочий проект	147	Ф 7.1-7.4
Впровадження	17	Д13
Всього	198	–

7.3 Визначення чисельності виконавців і планового фонду зарплати

Чисельність ставок інженерів-програмістів для розробки програмного забезпечення визначається за формулою:

$$Ч = \frac{T_{nz} N}{F_{pq} - H_{ев}}, \quad (7.5)$$

де: F_{pq} – плановий фонд робочого часу одного спеціаліста, днів;

T_{nz} – трудомісткість розробки програмного забезпечення люд-дні.

$$Ч = \frac{198 \cdot 1}{48 \cdot 5} = 4,6 \text{ ставки.}$$

Чисельність інженерів-електронщиків для проведення технічного обслуговування та ремонту комп'ютерних мереж визначається в залежності від наявності технічних засобів і норм витрат часу на виконання профілактичних робіт на протязі року.

Визначаємо затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за період розробки. Результати розрахунку зводимо до таблиці 7.3.

Таблиця 7.3 – Затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за розрахунковий період

Найменування обладнання	Профілактичне обслуговування			
	Кількість хв. на один. обл.	Кількість обладнання	Затрати часу в хв.	Затрати часу в год.
Системний блок ПК	385	12	4620	77
Монітор	160	12	1920	32
Клавіатура	140	12	1680	28
Маніпулятор «мишка»	30	12	360	6
Принтер матричний	185	1	185	3
Принтер лазерний	355	2	710	12
Принтер струминний	300	1	300	5
Сканер	155	2	310	5
Концентратор-маршрутизатор	155	2	310	5
Кабельні господарства ЛОМ на 1 м.п.	2,5	100	250	4
Кабельне господарство електромереж	48	50	2400	40
Копіювальний апарат	285	2	570	10
Усього за рік:			3 _ч	227

Час на профілактику обладнання в загальному балансі робочого часу інженерів-електронщиків не повинен складати більше 10%.

Виходячи з цього фонд робочого часу інженерів-електронщиків складає:

$$\Phi_{op}^c = \frac{3_{ч} \cdot n_{mic}}{1,2}, \quad (7.6)$$

$$\Phi_{op}^c = \frac{227 \cdot 2}{1,2} = 378,3 \text{ год.}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

$$Ч_{ел} = \frac{\Phi_{др}^c}{F_{др} \cdot T_{зм}}, \quad (7.7)$$

$$Ч_{ел} = 378,3/(48 \cdot 8) = 1 \text{ ставки.}$$

Для забезпечення нормального технічного обслуговування засобів ТО та мереж, необхідно прийняти найбільше ціле значення розрахункової чисельності інженерів-електронщиків.

Чисельність інженерів-системотехніків, адміністраторів мережі, дизайнерів WEB вузлів, системних програмістів (аналітиків), бухгалтерів-економістів визначається за потребою в залежності від функціональних обов'язків. Після визначення чисельності персоналу складається штатний розклад.

Таблиця 7.4 – Розрахунок чисельності штатного персоналу сектору системного та адміністративного обслуговування засобів ОТ та комп'ютерних мереж

Посада	Вид роботи	Час	К-ть штатних одиниць
Адміністратор загальної мережі, аналітик	Адміністрування локальної мережі, поштового та серверу DNS (OC FreeBSD), маршрутизатора Cisco, доменного контролеру Windows Server 2019, серверу доступу ADSL (OC Linux), налаштування ADSL, VPN PPPoE, Frame Relay, Wi-Fi	2	0,5
	Налаштування і конфігурування базової станції безпроводного зв'язку (CMTS)	0,5	
	Розробка та впровадження проектів з організації зв'язку між віддаленими об'єктами, ЛОМ	0,5	
	Забезпечення цілодобової роботи зв'язку клієнтів до мережі Інтернет	1	
Всього		4	

Продовження таблиці 7.4

Посада	Вид роботи	Час	К-ть штатних одиниць
Продакт-менеджер	Презентації нової продукції, пошук каналів збуту	1	0,25
	Підтримка постійних клієнтів	0,5	
	Оформлення договорів, ведення тендерів	0,25	
	Контроль взаєморозрахунків з постачальниками	0,25	
Всього		2	
Дизайнер WEB	Розробка концепції оформлення та інтерфейсу сайту, оптимізація дизайну існуючих, проектує їх структуру та навігацію	1	0,25
	Створення графічних і стилістичних елементів сайту	0,5	
	Оформлення банерів і промо-сторінок	0,25	
	Розміщення графіки і контенту на Інтернет сторінках	0,25	
Всього		2	
Інженер верстальник	Розробка та верстка макетів рекламної продукції та технічної документації	1	0,25
	Верстка друкованих видань	0,5	
	Додрукова підготовка макетів	0,25	
	Розміщення графіки і контенту на Інтернет сторінках	0,25	
Всього		2	

Складемо штатний розклад виконавців.

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньомісячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	1	16000	32000
Продакт-менеджер	0,25	13138	6569
Інженер-програміст	4,6	15000	138000
Інженер-електронщик	1	10000	20000
Інженер-системотехнік	0,25	10000	5000
Адміністратор мережі	0,5	10000	10000
Системний програміст	0,25	10000	5000
Дизайнер WEB	0,25	10000	5000
Інженер-верстальник	0,25	10000	5000
Бухгалтер-економіст	0,25	12500	6250
Всього за період розробки	$R_{cn} = 8,6$	-	$\Phi_{роб} = 232819$

Розрахуємо середньоденну зарплату одного виконавця:

$$z_{cd} = \frac{\Phi_{роб}}{R_{cn} F_{pq}}, \quad (7.8)$$

де: $\Phi_{роб}$ – загальна сума зарплати за плановий період, грн.

$$z_{cd} = \frac{232819}{8,6 \cdot 48} = 564 \text{ грн.}$$

7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника

Балансова вартість будівель визначається з урахуванням кількості робочих місць виконавців, питомої площі на одне робоче місце, та вартості одного квадратного метра виробничої площі:

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

$$B_{y\partial} = R_{cn}^1 S_y \Pi_{nl}, \quad (7.9)$$

де: R_{cn}^1 – кількість робочих місць виконавців, шт. Приймаємо 8 робочих місць;

S_y – питома площа на одне робоче місце, m^2 ;

Π_{nl} – вартість одного квадратного метра площі, грн.

Згідно даних ТОВ науково-дослідницького консалтингового підприємства «Пектораль» (м. Кіровоград, вул. Глинки 16) ціна одного квадратного метра площі новобудови, вік якої не перевищує 25 років, по місту складає 500...1600 у.о./ m^2 . Враховуючи, що курс складає 1 у.о. = 37 грн. приймаємо для розрахунку вартість одного метра квадратного рівною 20000 грн./ m^2 . На кожне робоче місце у середньому потрібно 8 m^2 . З урахуванням цього:

$$B_{y\partial} = 9 \cdot 8 \cdot 20000 = 1440000 \text{ грн.}$$

Вартість передавальних пристроїв складає 10% від вартості будівель, і у даному випадку вона складе: 144000 грн.

Балансова вартість інвентарю розраховується за нормою 3500 грн. на одне робоче місце. Тобто:

$$I_{не} = R_{cn}^1 \cdot \Pi_m, \quad (7.10)$$

де: Π_m – ціна меблів для одного робочого місця, грн.

$$I_{не} = 9 \cdot 3500 = 31500 \text{ грн.}$$

Балансова вартість обчислювальної техніки визначається по оптовим цінам постачальника з врахуванням витрат на транспортування.

Специфікація на обчислювальну техніку наведена в таблиці 7.7.

Дані по оптовій ціні на обладнання та комплектуючі вибирались по прайсу фірми supercomp.kiev.ua за 29.10.23 – джерело <http://supercomp.kiev.ua>.

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Персональний комп'ютер		12721
Системний блок		7721
Процесор	AMD Ryzen 5 4500 (100-000000644) AM4, ядер, 12 потоків, 3.6 GHz, 4.1 GHz, TDP – 6 Вт, 7nm, L1: 384KB, L2: 3MB, L3: 8MB	-
Системна плата	ASUS PRIME A320M-K AM4, DDR4, 3200 MHz, LAN – 1 Гбіт/с, D-Sub (VGA), HDMI 1 x M.2, 4 x SATA 6.0 Gb/s, Micro-ATX	-
Відеокарта	Radeon R7 350 2Gb Afox (AFR7350 2048D5H4-V3) PCI-Express 3.0, 2 GB GDDR5, 128 Bit	-
Жорсткий диск	SSD 2.5" 256GB Mibran (MI2.5SSD/CA256GBST) 256 GB, 3D TLC NAND, 2.5", SATA III (6Gb/s)	-
Оперативна пам'ять	DDR4 8GB 3200 MHz Dat (DT8G4DLND32) DDR4, 8 ГБ, В наборі 1 шт	-
DVD-привод	-	-
Корпус	Vinga CS105B-450W, Miditower, ATX Micro – ATX, Mini – ITX, PSU – 450 Вт	
Кардридер внутрішній	Transcend TS-RDF8K USB 3.0	
інше	Клавіатура, мишка	Подаруно

Продовження таблиці 7.5

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Монітор	Монітор BenQ GL2450HM Black	3600
Принтер лазерний	Canon i-SENSYS LBP6030W	2700
Принтер струминний	Epson Stylus Photo P50 (C11CA45341) + USB cable	5500
Сканер	Epson Perfection V37	2800
Копіювальний апарат	Canon i-SENSYS MF217W with Wi-Fi	5965
Пристрій безперебійного живлення	Powercom BNT-600AP USB	1400

Витрати на транспорт, монтаж та випробування можуть бути прийняті в межах до 10% від оптової ціни.

Для визначення необхідної кількості капітальних вкладень складом таблицю 7.8.

Таблиця 7.7 – Балансова вартість обчислювальної техніки

Найменування обчислювальної техніки	Кількість, шт.	Ціна за одиницю, грн.	Витрати на транспортування, монтаж та випробування.	Загальна вартість, грн.
Персональні комп'ютери	9	12721	11448,9	125937,9
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Сканери	1	2800	280	3080
Копіюв. апарат	1	5965	596,5	6561,5
Всього	—	—	—	147569,4

Таблиця 7.8 – Вартість основних фондів та амортизаційні відрахування розробника

Групи та види основних фондів	Балансова вартість, грн.	Амортизація	
		Норма, %	Відрахування, грн.
1	2	3	4
Група 3			
1. Будівлі	1440000	-	-
2. Передавальні пристрої	144000	-	-
Всього по групі	1584000	5	79200
Група 4			
3. Обчислювальна техніка	147570	-	-
Всього по групі	147570	40	59028
Нематеріальні активи			
4. Нематеріальні активи	17000	10	1700
Група 5, 6			
5. Вимірювальні пристрої	9031	25	2257,75
6. Транспортні засоби	121875	20	24375
7. Господарський інвентар	31500	25	7875
Всього по групі	162406	-	34507,75
Разом	$K_p = 1910976$		$A_p = 174435,75$

Примітка: вартість автомобіля ГАЗ Газель взята по даним електронного ресурсу <http://www.avtopoisk.ua>, що враховуючи курс 37 складає 121875 грн.

Згідно виданих викладачем норм приймаємо 0,5 пачки паперу на місяць розробки. Тоді, враховуючи, що вартість пачки паперу складає $C_n = 70$ грн., визначаємо вартість паперу за період розробки $N_m = 2$ міс:

$$Z_{M1} = C_n \cdot N_m. \quad (7.16)$$

$$Z_{M1} = 0,5 \cdot 70 \cdot 2 = 70 \text{ грн.}$$

Згідно норм до вартості запам'ятовуваних пристроїв входить вартість CD дисків за потребою та одного DVD диска для збереження резервної копії програми:

$$Z_{M2} = \sum C_d, \quad (7.17)$$

де: C_d – вартість дисків CD/DVD: CDR TDK 700Mb, 80Min, 52x Cake box – 22 грн./шт., DVD-R LG 4,7Gb, 16x speed Cake box – 36 грн./шт.

$$Z_{M2} = 36 \cdot 1 = 36 \text{ грн.}$$

Згідно виданих викладачем норм одноразовій заправці підлягають усі друкуючі пристрої і становить:

$$Z_{M3} = \sum C_z, \quad (7.18)$$

де: C_z – вартість розхідних матеріалів друкуючих пристроїв: відновлення та заправка картриджу для Canon i-SENSYS LBP6030W – 574 грн.; картридж для Epson Stylus Photo P50 – 558 грн.; відновлення картриджу для MF217W – 570 грн.

$$Z_{M3} = 500 + 508 + 1155 = 2163 \text{ грн.}$$

$$Z_M = (70 + 36 + 2163) / 17 = 133 \text{ грн.}$$

Визначимо витрати на освоєння нових мов програмування або операційних систем за нормативом ($H_n = 15\%$) від основної зарплати виконавців:

$$O_n = Z_o \cdot H_n \cdot 0,01, \quad (7.19)$$

де: H_n – норматив витрат на освоєння нових мов програмування, %.

$$O_n = 6579 \cdot 15 \cdot 0,01 = 987 \text{ грн.}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_e = 17$ прим.):

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

$$A_m = \frac{A_p \cdot N_{mic}}{N_e \cdot 12}, \quad (7.20)$$

де: A_p – загальна річна сума амортизаційних відрахувань, грн.

$$A_m = 174436 \cdot 2 / (17 \cdot 12) = 1710 \text{ грн.}$$

Величини ціна підприємства, податок на додану вартість, відпускна ціна програмної продукції визначаються за формулами, приведеними в таблиці 7.9

Таблиця 7.9 – Нормативна калькуляція собівартості розробки програмного забезпечення задачі

Найменування статей витрат	Позначення	Величина, грн
1	2	3
1. Основна зарплата виконавців	Z_o	6579
2. Додаткова зарплата виконавців	Z_o	658
3. Відрахування на соціальні потреби	C_{oc}	2678
4. Загальногосподарські витрати	G_{ocn}	987
5. Витрати на матеріали	Z_M	133
6. Освоєння нових операційних систем, мов програмування	O_n	987
7. Амортизація основних фондів	A_m	1710
8. Повна собівартість програмного забезпечення	C_n	13732
9. Плановий прибуток	P_p	6866
10. Ціна підприємства $C_n = C_n + P_p$	C_n	20598
11. Податок на додану вартість $ПДВ = 0.01 \cdot H_{об} \cdot C_n$	$ПДВ$	4119,6
12. Відпускна ціна програмної продукції $C = C_n + ПДВ$	C	24717,6

Повна собівартість ПЗ визначається як сума витрат за попередніми статтями калькуляції:

$$C_n = Z_o + Z_d + C_{oc} + \Gamma_{ocn} + Z_m + O_n + A_m. \quad (7.21)$$

$$C_n = 6579 + 658 + 2678 + 987 + 133 + 987 + 1710 = 13732 \text{ грн.}$$

Визначимо плановий прибуток за рівнем рентабельності (P_n) програмної продукції, яка залежить від складності програми та ступеня новизни задачі.

Для даного програмного забезпечення рівень рентабельності складає 50%.

$$P_p = 0,01 \cdot P_n \cdot C_n, \quad (7.22)$$

де: P_n – рівень рентабельності, %.

$$P_p = 0,01 \cdot 50 \cdot 13732 = 6866 \text{ грн.}$$

7.6 Визначення об'єму капітальних вкладень у споживача програмної продукції

Об'єм капітальних вкладень у споживача програмної продукції визначаємо на основі балансової вартості основних фондів, яка враховує ціну, транспортно-заготівельні витрати, вартість будівель, монтажних та пусконаладжувальних робіт, а також витрати на випробування у виробничих умовах.

Таблиця 7.10 – Розрахунок об'єму капітальних вкладень у споживача програмної продукції

Найменування капітальних вкладень	Сума за варіантами, грн.	
	Базовий	Новий
Вартість програмної продукції	–	24718
Всього капітальних витрат	–	24718

7.7 Визначення експлуатаційних витрат

Експлуатаційні витрати у споживача програмної продукції визначаємо при умові роботи підсистеми на протязі року. Результати зводимо до таблиці 7.11.

Таблиця 7.11 – Розрахунок експлуатаційних витрат у споживача програмної продукції

Найменування статей витрат	Позначення	Сума витрат за варіантами, грн.	
		Базовий	Новий
1. Витрати на обслуговування	Z_p	26572	11810
2. Витрати на електроенергію	$Z_{ел}$	205	137
3. Витрати на амортизацію	$Z_{ам}$	0	6180
Всього витрат за рік	I	26777	18127

Витрати на обслуговування системи:

$$Z_p = T_p \cdot Z_z \cdot (1 + 0,01 \cdot H_q) \cdot (1 + 0,01 \cdot H_c), \quad (7.23)$$

де: T_p – кількість годин обслуговування за рік, год.;

Z_z – заробітна плата обслуговуючого персоналу, грн/год.

Після купівлі нового програмного забезпечення кількість профілактичних годин робіт зменшилася з 180 годин на рік до 80 годин на рік, тому витрати на технічне обслуговування зменшилися з:

$$Z_{p \text{ баз}} = 180 \cdot 110 \cdot 1,1 \cdot 1,22 = 26572 \text{ грн},$$

до:

$$Z_{p \text{ нов}} = 80 \cdot 110 \cdot 1,1 \cdot 1,22 = 11810 \text{ грн}.$$

Витрати на електроенергію визначаються з урахуванням споживаємої потужності ($P_{ел}$) в кіловатах, часу експлуатації технічних засобів (T_p) в годинах та ціни однієї кіловат-години ($C_{ел}$):

$$Z_{ел} = P_{ел} \cdot T_p \cdot C_{ел}. \quad (7.24)$$

$$Z_{ел \text{ баз}} = 0,45 \cdot 120 \cdot 3,8 = 205 \text{ грн}.$$

$$Z_{ел \text{ нов}} = 0,45 \cdot 80 \cdot 3,8 = 137 \text{ грн}.$$

Витрати по амортизації визначаються на основі норм амортизаційних відрахувань, вартості програмної продукції і основних фондів. Для розрахунку складаємо таблицю 7.12.

Таблиця 7.12 – Розрахунок амортизаційних відрахувань

Групи основних фондів	Норма амортизації %	Балансова вартість, грн., за варіантами		Сума відрахувань, грн за варіантами	
		Базовий	Новий	Базовий	Новий
Програмна продукція	25	–	24718	–	6179,5
Всього відрахувань	-	–	24718	–	6179,5

7.8 Визначення економічної ефективності програмної продукції

Економічна ефективність програмного забезпечення визначається для виготовлювача і споживача за такими показниками.

Величина економічного ефекту при виготовленні програмної продукції, розраховуємо за формулою:

$$E_e = (C_n - C_n) \cdot N_e - \sum_{i=1}^m E_{p_m} \cdot K_{p_m}, \quad (7.24)$$

де: K_p – балансова вартість основних фондів розробника, грн.; E_p – розрахунковий коефіцієнт капіталовкладень.

$$E_e = (20598 - 13732) \cdot 17 - (0,05 \cdot 1584000 + 0,4 \cdot 147570 + 0,2 \cdot 121875 + 0,25 \cdot 40531 + 0,1 \cdot 17000) \cdot 2/12 = 87649 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у виробника програмної продукції:

$$T_e = \frac{K_p^*}{(C_n - C_n) \cdot N_e}, \quad (7.25)$$

де: K_p^* – балансова вартість основних фондів розробника без врахування вартості ОФ третьої групи, так як їх строк служби на порядок більший ніж період розробки ПЗ.

$$T_e = \frac{326976}{(20598 - 13732) \cdot 17 \cdot 12 / 2} = 0,5 \text{ років.}$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1. Кількість екземплярів програми	Прим.	17
2. Повна собівартість розробленої програми	Грн.	13732
3. Ціна розробленої програми	Грн.	20589
4. Плановий прибуток від реалізації розробленої програми	Грн.	6857
5. Рентабельність програмної продукції	%	50
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	1910976
7. Загальний прибуток від реалізації програмної продукції	Грн.	116569
8. Величина економічного ефекту при виготовлені програмної продукції	Грн.	87649
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років	0,5
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	24718
11. Величина економічного ефекту у користувача програмної продукції	Грн.	2471
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	2,86

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

$$E_{cn} = (I_{\bar{o}} - I_n) - E_n(K_n - K_{\bar{o}}), \quad (7.27)$$

де: $I_{\bar{o}}$, I_n – величина експлуатаційних витрат за базовим и новим варіантом відповідно;

$K_{\bar{o}}$, K_n – об'єм капітальних вкладень за варіантами, що порівнюються.

$$E_{cn} = (26777 - 18127) - 0,25 \cdot 24718 = 2471 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у споживача програмної продукції за рахунок зниження експлуатаційних витрат:

$$T_{cn} = \frac{K_n - K_{\bar{o}}}{I_{\bar{o}} - I_n}, \quad (7.28)$$

$$T_{cn} = \frac{24718}{26777 - 18127} = 2,86 \text{ року.}$$

7.9 Висновки

Розроблена програма економічно вигідна. За рахунок впровадження програмного забезпечення досягається скорочення часу обробки інформації, підвищується культура праці, підвищення якості приймаючих управлінських рішень.

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Охорона здоров'я працівників, забезпечення безпеки умов праці, ліквідація професійних захворювань і виробничого травматизму повинна складати одну з головних завдань роботодавця.

Основою охорони праці є науковий аналіз умов праці, технологічних процесів, виробничого обладнання, робочих місць, трудових операцій, організації виробництва з метою виявлення шкідливих і небезпечних виробничих факторів, їх властивостей, особливостей впливу на організм людини. На підставі такого аналізу розробляються заходи та засоби, спрямовані на мінімізацію несприятливого впливу виробничих факторів, створення безпечних та нешкідливих умов праці. Основою охорони праці є науковий аналіз умов праці, технологічних процесів, виробничого обладнання, робочих місць, трудових операцій, організації виробництва з метою виявлення шкідливих і небезпечних виробничих факторів, їх властивостей, особливостей впливу на організм людини. На підставі такого аналізу розробляються заходи та засоби, спрямовані на мінімізацію несприятливого впливу виробничих факторів, створення безпечних та нешкідливих умов праці. Для того, щоб об'єктивно проаналізувати відповідність умов праці діючим нормативно-правовим актам, необхідно здійснити санітарно-гігієнічну характеристику умов праці відділу, в якому працює програміст, над розробкою даного програмного продукту. В зв'язку з цим необхідно сконцентрувати увагу на небезпечних і шкідливих чинниках пов'язаних з постійною роботою за комп'ютером.

Електробезпека є одним із критичних питань для співробітників, що працюють із технікою, яка одержує живлення з електричної мережі. При невиконанні норм електробезпеки можлива поразка електричним струмом.

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

Електронно-обчислювальна машина (ЕОМ) та інше обладнання є джерелами небезпеки ураження електричним струмом. Так як робота програміста характеризується істотним зоровим навантаженням, то вимагає належного освітлення. У приміщенні, в якому працюють люди (у т.ч. програмісти) необхідно створити належний мікроклімат, параметри якого регламентуються, Державними санітарними правилами і нормами, зокрема ДСанПіН 3.3.2.007-98.

При роботі з використанням ЕОМ відзначають наступні небезпечні та шкідливі фактори:

- ризик виникнення надзвичайних ситуацій природного або штучного характеру на об'єкті або території.
- ризик виникнення пожежі;
- негативний вплив на органи зору людини;
- ризики ураження електричним струмом;
- недостатня, або надмірна освітленість робочого місця;
- електромагнітні (у т.ч. високочастотні) електромагнітні випромінювання (коливання);
- несприятливі мікрокліматичні умови;
- нервово-емоційна напруженість праці;
- інтелектуальні навантаження;
- монотонність праці;
- невідповідність ергономічних показників робочого місця діючим вимогам;
- шум;
- статичні навантаження на кістково-м'язовий апарат;

Відповідно до ст.14 Закону «Про охорони праці» [1] на роботодавця покладено обов'язок забезпечити: безпеку працівників при експлуатації устаткування; застосування коштів індивідуальної захисту працівників;

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

відповідні вимоги охорони праці, умови праці в кожному робоче місце; дотримання режиму праці та відпочинку працівників; навчання безпечним методам і прийомам виконання; інструктаж з охорони праці; організацію контролю над станом умов праці в робочих місць; проведення атестації робочих місць в умовах праці.

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

8.3 Розробка заходів з умов поліпшення охорони праці

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

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

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

Рекомендовані заходи: регулярні періодичні наочні огляди персоналом шляхів для евакуації людей із приміщення, відповідно до плану евакуації (який повинен розташовуватись на видному місці у приміщенні), включення до колективного договору мінімально можливого вмісту аптечок з обов'язково наявністю масок-клапанів, або іншого спорядження для штучного дихання. Регулярна періодична перевірка параметрів заземлення та занулення (вимірювання опору ланцюга) [9].

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

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

Так як при ураженні електричним струмом у людини може статися фібриляція шлуночків серця, в організації бажано мати дефібрилятор і підготовлений персонал для роботи з ним.

8.4 Пожежна безпека

Вимоги до пожежної безпеки на підприємстві неухильно повинен дотримуватися кожен співробітник, а організаційна складова при цьому покладається на посадових осіб за відповідним рішенням керівництва і прописується в посадових інструкціях і положеннях по структурним підрозділам.

Зокрема, вказуються конкретні території, ділянки, зони, об'єкти, цілі будівлі і їх частини, поверхи, на яких відповідального співробітника повинне проводити такі організаційні роботи.

Відповідальні особи зобов'язуються розробити, впровадити та підтримувати в певному інструкцією і положенням на ввірених їм об'єктах протипожежний режим і інструкції відповідно до вимог, викладених в нормативних актах.

Передбачено також створення підрозділу добровільної пожежної охорони та пожежно-рятувальної команди в його складі.

Встановлений режим включає порядки з описом місць спеціального призначення та правила їх користування та утримання, наприклад:

- евакуаційних шляхів;
- так званих «курилок»;
- місць складування продукції та сировини;
- стоянки транспорту.

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		94

Також встановлюється порядок роботи та технічного обслуговування:

- вентиляційного устаткування;
- засобів пожежогасіння і захисту від загорянь;
- нагрівальних приладів;
- електрообладнання.

Розробляються і впроваджуються правила роботи з відкритим вогнем і горючими матеріалами. Створюються графіки проходження інструктажів з пожежної безпеки співробітників, а також порядок і терміни перевірок знань пожежно-технічного мінімуму, в тому числі, тих працівників, які відповідальні за цю ділянку роботи на підприємстві. При цьому можуть передбачатися внутрішні лекції, семінари, тренінги та практичні заняття на підприємстві, а також зовнішні – на базі спеціалізованих навчальних центрів з професійними викладачами.

Важливою складовою протипожежного режиму на будь-якому об'єкті є розробка і впровадження порядку дій при виникненні пожежі. Неодмінно має бути план евакуації, описано, як повинні відключатися електроустановки, що і в якій послідовності необхідно робити співробітникам.

Відповідно, для кожного об'єкта, кожного приміщення (крім коридорів, санвузлів, басейнів і подібних приміщень), окремих видів робіт складаються інструкції, за якими повинен працювати персонал, залучений на певних ділянках і в виконанні окремих видів робіт. За інструкціями проводиться навчання (інструктаж) персоналу з подальшим контролем знань.

Детально про те, як розробити протипожежний режим, прописати порядки та інструкції, пояснюють на тематичних курсах і семінарах. [4]

Відповідно ДБН В 1.1-7-2016 «Пожежна безпека об'єктів будівництва» будинок можна віднести до II групи по ступені вогнестійкості й до категорії Д по ступені пожежонебезпеки.

Від розподільного щита по праву й ліву сторони встановлені кондиціонери, зовнішня електропроводка, поміщена в ізолюваний кабель. Висота

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		95

проводки становить 2,2 м від рівня підлоги, її кріплення здійснюється за допомогою металевих власників. Біля кожного стола організований розподільний щит, розташований на текстолітовій пластинці, закріпленої на стіні на рівні 1 м від підлоги. Усього до складу входять п'ять розеток і дві клеми заземлення. Всі обчислювальні машини з'єднані із клемми заземлення. Чотири з п'яти розеток забезпечують подачу напруги 220 V, а одна, забезпечує подачу напруги в 36 V. Про це є відповідні написи на кожному розподільному щиті.

Робота обслуговуючого персоналу полягає в інсталяції необхідного програмного забезпечення й наступному його використанні в діалоговому режимі роботи з ЕОМ. Іноді може виникати необхідність написання допоміжних програм для поліпшення роботи вузла або для зниження витрат. З погляду забезпечення умов праці й вимог техніки безпеки для роботи програміста необхідно наступне: достатнє висвітлення екрана дисплея й робочого місця; повна технічна справність устаткування, його електробезпечність; достатня пожежна безпечність приміщення; оптимальний мікроклімат, що сприяє продуктивній роботі; відповідність робочого місця вимогам ергономіки. До небезпечних і шкідливих факторів, дії яких піддається програміст, можна віднести: можливість поразки електричним струмом, при електроні справності встаткування, порушенні заземлення або техніки безпеки; робота в мікрокліматі з неприпустимими параметрами; робота при недостатній освітленості екрана дисплея й робочого місця.

Відповідно НПАОП 40.1-1.21-98 “Правил безпечної експлуатації електроустановок споживачів” [6], приміщення можна віднести до приміщень без підвищеної небезпеки, оскільки це приміщення, сухе, з нормальною температурою й ізолюючими підлогами, що не має заземлених металоконструкцій.

Персональні ЕОМ можна віднести до першого класу електротехнічних виробів по способі захисту людини від поразки електричним струмом, оскільки їхні корпуси зроблені з ізолюючої пластмаси й кожен пристрій має заземлення.

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		96

Відповідно правилам пристрою електроустановок ЕОМ можна віднести до електроустановок з робочою напругою до 1000 В.

Однієї з достовірних причин пожежі в приміщенні з обчислювальною технікою може бути коротке замикання, що спричиняє спалах електропроводки. Для його попередження вся обчислювальна техніка, а також інші електричні пристрої повинні бути обладнані плавкими запобіжниками, а на вході електромережі повинен бути передбачений автомат захисту. Не слід користуватися електричними подовжувачами й трійниками, що не мають сертифікатів відповідності вимогам безпеки.

Необхідно передбачити наявність у межах досяжності первинних засобів гасіння пожежі (вогнегасників) для локалізації вогню власними засобами до приїзду команди пожежної охорони. Повинен бути розроблений план екстреної евакуації персоналу при виникненні загоряння. Кількість евакуаційних виходів повинне бути не менш двох. Допускається використання одного евакуаційного виходу, якщо відстань найбільш віддаленого робочого місця до цього виходу не перевищує 25 м.

8.5 Розрахункова частина

Запорукою безпечної роботи є виконання вимог електричної безпеки, оскільки все офісне обладнання заживлюється від електричної мережі. Одним з необхідних засобів електричної безпеки є встановлення захисного заземлення.

Початкові дані, необхідні для розрахунку захисного заземлення:

- допустимий опір розповсюдженню струму в землі від заземлювального пристрою $R_{zn} = 10 \text{ Ом}$;
- питомий опір ґрунту в місці встановлення заземлювача $\rho_3 = 100 \text{ Ом/м}$;
- тип ґрунту – суглинок;
- тип заземлювача – труба, діаметром $d=0.058 \text{ м}$ і довжиною $l = 2.1 \text{ м}$;
- конструкція заземлювача – розташування заземлювачів по контуру. Розрахунок

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

9. Методичні рекомендації до виконання розділу "Заходи з охорони праці та техніки безпеки" випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти для здобувачів вищої освіти спеціальностей 123 "Комп'ютерна інженерія" та 122 "Комп'ютерні науки" / М-во освіти і науки України, Центральнoукраїн. нац. техн. ун-т, каф. кібербезпеки та програм. забезпечення; [укл. О.В. Оришака, К.М. Марченко]. – Кропивницький: ЦНТУ, 2022. – 19 с. [Електронний ресурс]. – Режим доступу: <http://dspace.kntu.kr.ua/jspui/handle/123456789/12240> (дата звернення 19.09.22).

4. Зеркалов Д. В. Охорона праці в Галузі: Загальні вимоги: навч. посіб. Київ: Основа. 2011. 551 с.

6. НПАОП 40.1-1.21-98. Правила безпечної експлуатації електроустановок споживачів – Режим доступу до ресурсу: https://dnaop.com/html/2029/doc_40.1-1.21-НПАОА_40.1-1.21-98 (дата звернення 19.09.23).

КБПЗ – 2023

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		100

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи моніторингу центрів обробки даних, побудованих з використанням мереж NGBaseT.

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 моніторингу центрів обробки даних, побудованих з використанням мереж NGBaseT.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем моніторингу центрів обробки даних, побудованих з використанням мереж NGBaseT.
- Досліджена система моніторингу центрів обробки даних, побудованих з використанням мереж NGBaseT.
- На основі отриманих результатів досліджень створена програмна реалізація системи моніторингу центрів обробки даних, побудованих з використанням мереж NGBaseT.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання моніторингу центрів обробки даних, побудованих з використанням мереж NGBaseT.

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-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		101

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 Delphi 10.4.1. 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 DES.

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.

Розроблена програма має реальний економічний ефект від її впровадження у виробництво у сумі 2471 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 2,86 роки.

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		102

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Маляренко Р.В. Дослідження та програмна реалізація системи моніторингу центрів обробки даних, побудованих з використанням мереж NGBaseT // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2023.
2. 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.
3. 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.
4. 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.
5. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». *CEUR Workshop Proceedings*. Volume 2740, 2020, Pages 102-114.
6. 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.
7. 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.

8. 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.

9. 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.

10. 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.

11. 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.

12. 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.

13. 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.

14. 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.

15. Smirnov, O., Odarchenko, R., Abakumova, A., Usik, P., Kundyzy, 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.

16. 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.

17. 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.

18. 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.

19. 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.

20. 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.

21. 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.

					БКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		105

22. 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.

23. 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.

24. 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.

25. 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.

26. 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.

27. 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

28. 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.

29. 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.

30. 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.

31. 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.

32. 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.

33. 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.

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

35. 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

36. Дреєва Г.М., Смірнов О.А., Дреєв О.М. Метод генерування фрактальноподібної числової послідовності на основі скінченного автомату для

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		107

модельовання трафіку у мережі. Центральнуукраїнський науковий вісник. Технічні науки. № 1(32). с. 173-183, 2019.

37. 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.

38. 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.

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

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

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

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

43. Смірнов О.А., Смірнов С.А., Дідик А.К., Дреєв А.М. Спосіб контролю ліній зв'язку телекомунікаційної системи антивірусу. Спосіб контролю ліній зв'язку телекомунікаційної системи антивірусу. Збірник наукових праць

Харківського університету Повітряних Сил. Випуск 2 (47). – Харків: ХУПС. – 2016. – С. 121-127.

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

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

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

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

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

49. Смірнов О.А., Стасєв Ю.В. Бараннік В.В. Захист інформації в автоматизованих системах управління. Навчальний посібник – Харків: ХУПС, 2015. – 264 с.

50. Смірнов О.А., Коваленко О.В., Кожанова А.С., Лєвошко О.Л., Константинова Л.В. Основи системного програмування. Навчальний посібник для студентів вищих навчальних закладів напрямів підготовки 8.050102 «Комп'ютерна інженерія». За ред. Коваленка О.В., Гриф «Навчальний посібник» надано у відповідності з листом Міністерства освіти і науки України від 26.02.2013 року № 1/11-4368. – Кіровоград: КНТУ 2013. – 257с.

51. Смірнов О.А., Мелешко Є.В., Семенов С.Г. Методи та засоби обробки сигналів і даних в інформаційних системах. Навчальний посібник для студентів вищих навчальних закладів напрямів підготовки 8.050102 «Комп'ютерна

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		109

інженерія». За ред. О.А. Смірнова Гриф “Навчальний посібник” надано у відповідності з листом Міністерства освіти і науки, молоді та спорту України від 17.04.2012 року № 1/11-5249. – Кіровоград: КНТУ 2012. – 250 с.

52. Смірнов О.А., Євсєєв С.П., Жукарев В.Ю., Король О.Г., Сорокін В.Є., Мелешко Є.В. Технології і стандарти комп’ютерних мереж. Навчальний посібник для студентів вищих навчальних закладів напрямів підготовки 8.050102 «Комп’ютерна інженерія» та 8.0925 «Автоматизація й комп’ютерно-інтегровані технології». За ред. О.А. Смірнова Гриф “Навчальний посібник” надано у відповідності з листом Міністерства освіти і науки, молоді та спорту України від 1.12.2011 року № 1/11-11258. – Кіровоград: КНТУ 2012. – 454 с

КБПЗ – 2023

					ВКРМ-122.23.0079.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		110

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-122.23.0079.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Маляренко Р.В.				<i>Дослідження та програмна реалізація системи моніторингу центрів обробки даних, побудованих з використанням мереж NGBaseT</i>	Літ.	Аркуш	Аркушів
Перевірів	Лисенко І.А.					М	1	6
Н. Контр.	Коваленко А.С.					ЦНТУ КН-22МЗ		
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи моніторингу центрів обробки даних, побудованих з використанням мереж NGBaseT.

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. № 37-13 від 04.08.2023 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи моніторингу центрів обробки даних, побудованих з використанням мереж NGBaseT.

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-122.23.0079.00.00.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 моніторингу центрів обробки даних, побудованих з використанням мереж NGBaseT;
- 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-122.23.0079.00.00.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 архітектури IBM PC, 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 Delphi 10.4.1.

					ВКРМ-122.23.0079.00.00.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 розрахунок витрат показників економічного ефекту з урахуванням цін на 3 вересня 2023 року.

8 Вимоги щодо охорони праці

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

					ВКРМ-122.23.0079.00.00.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 – 110 аркушів.

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 Розрахунок з техніко-економічного обґрунтування.

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 10.12.2023 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 14.12.2023 р.

					ВКРМ-122.23.0079.00.00.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

_____ Лисенко І.А

*Дослідження та програмна реалізація
системи моніторингу центрів обробки даних, побудованих з використанням
мереж NGBaseT*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 49

Літера: РП

Кропивницький – 2023 року

Основна програма**Файл Monitoring_NGBaseT.dpr основної програми**

```
program Monitoring_NGBaseT;

uses
  Forms,
  Main in `Main.pas' {MainForm},
  About in `About.pas' {Form1},
  TCP_IP in `TCP_IP.pas' {Form2},
  Stat in `Stat.pas' {Form3};

{$R *.res}

begin
  Application.Initialize;
  Application.CreateForm(TMainForm, MainForm);
  Application.CreateForm(TForm1, Form1);
  Application.CreateForm(TForm2, Form2);
  Application.CreateForm(TForm3, Form3);
  Application.Run;
end.
```

КБПЗ_2023

Файл Main.pas основної програми

```

unit Main;

interface

// опис бібліотек

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, ComCtrls, Stat,
  ShellAPI, ShlObj, ImgList, TCP_IP, About;

//опис типів

type
  TMainForm = class(TForm)
    gbxShares: TGroupBox;
    lbxShares: TListBox;
    gbxSessions: TGroupBox;
    lvSessions: TListView;
    bvlSessions: TBevel;
    gbxFiles: TGroupBox;
    btnGetShares: TButton;
    btnCloseShares: TButton;
    btnAddShares: TButton;
    btnCloseSession: TButton;
    btnGetSessions: TButton;
    bvlTopSessions: TBevel;
    plButtonFiles: TPanel;
    btnGetFiles: TButton;
    btnCloseFile: TButton;
    bvlLeftFiles: TBevel;
    plFiles: TPanel;
    lvFiles: TListView;
    bvlTopFiles: TBevel;
    gbxTraffic: TGroupBox;
    lvTraffic: TListView;
    bvlTraffic: TBevel;
    tmrTraffic: TTimer;
    Button1: TButton;
    rgScope: TRadioGroup;
    GroupBox1: TGroupBox;
    cbUsageAll: TCheckBox;
    cbUsageConnectable: TCheckBox;
    cbUsageContainer: TCheckBox;
    GroupBox2: TGroupBox;
    cbTypeAny: TCheckBox;
    cbTypeDisk: TCheckBox;
    cbTypePrint: TCheckBox;
    NetTree: TTreeView;
    ImageList1: TImageList;
    Button2: TButton;
    Button3: TButton;
    Button4: TButton;
    function IsNT(var Value: Boolean): Boolean;
    procedure btnGetSharesClick(Sender: TObject);
    procedure btnCloseSharesClick(Sender: TObject);
    function SelectDirectory: String;
    procedure btnAddSharesClick(Sender: TObject);
    function CardinalToTimeStr(Value: Cardinal):String;
    procedure btnGetSessionsClick(Sender: TObject);
    procedure btnCloseSessionClick(Sender: TObject);
    procedure btnGetFilesClick(Sender: TObject);
    procedure btnCloseFileClick(Sender: TObject);
    procedure tmrTrafficTimer(Sender: TObject);
    procedure Button1Click(Sender: TObject);
  end;

```

```

procedure NetTreeCustomDrawItem(Sender: TCustomTreeView;
  Node: TTreeNode; State: TCustomDrawState; var DefaultDraw: Boolean);
procedure NetTreeDbClick(Sender: TObject);
procedure NetTreeGetImageIndex(Sender: TObject; Node: TTreeNode);
procedure Button4Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject);

//опис типів та записів

private
  { Private declarations }
public
  { Public declarations }
  SessionCloseKey: array [0..512] of SmallInt;
  procedure Open_Do_Close_Enum(const ParentNode: TTreeNode;
    ResScope, ResType, ResUsage: DWORD; const NetContainerToOpen:
PNetResource);
  // function OpenEnum(const NetContainerToOpen: PNetResource;
  //   ResScope, ResType, ResUsage: DWORD): THandle;
  // function EnumResources(const ParentNode: TTreeNode;
  //   ResScope, ResType, ResUsage: DWORD; hNetEnum: THandle): UINT;
  end;

type
  TShareInfo2 = packed record
    shi2_netname : PWChar;
    shi2_type: DWORD;
    shi2_remark :PWChar;
    shi2_permissions: DWORD;
    shi2_max_uses : DWORD;
    shi2_current_uses : DWORD;
    shi2_path : PWChar;
    shi2_passwd : PWChar;
  end;
  PShareInfo2 = ^ TShareInfo2;
  TShareInfo2Array = array [0..512] of TShareInfo2;
  PShareInfo2Array = ^ TShareInfo2Array;

type
  TShareInfo50 = packed record
    shi50_netname : array [0..12] of Char;
    shi50_type : Byte;
    shi50_flags : Word;
    shi50_remark : PChar;
    shi50_path : PChar;
    shi50_rw_password : array [0..8] of Char;
    shi50_ro_password : array [0..8] of Char;
  end;

type
  TSessionInfo502 = packed record
    Sesi502_cname: PWideChar;
    Sesi502_username: PWideChar;
    Sesi502_num_opens: DWORD;
    Sesi502_time: DWORD;
    Sesi502_idle_time: DWORD;
    Sesi502_user_flags: DWORD;
    Sesi502_cltype_name: PWideChar;
    Sesi502_transport: PWideChar;
  End;
  PSessionInfo502 = ^TSessionInfo502;
  TSessionInfo502Array = array[0..512] of TSessionInfo502;
  PSessionInfo502Array = ^TSessionInfo502Array;

type
  TSessionInfo50 = packed record
    Sesi50_cname : PChar;

```

```

    Sesi50_username      : PChar;
    sesi50_key           : Cardinal;
    sesi50_num_conns    : Word;
    sesi50_num_opens    : Word;
    sesi50_time          : Cardinal;
    sesi50_idle_time    : Cardinal;
    sesi50_protocol     : Byte;
    pad1                 : Byte;
end;

```

```
type
```

```

TFileInfo3 = packed record
    fi3_id             : DWORD;
    fi3_permissions   : DWORD;
    fi3_num_locks     : DWORD;
    fi3_pathname      : PWChar;
    fi3_username      : PWChar;
end;
PFileInfo3 = ^TFileInfo3;
TFileInfo3Array = array[0..512] of TFileInfo3;
PFileInfo3Array = ^TFileInfo3Array;

```

```
type
```

```

TFileInfo50 = packed record
    fi50_id           : Cardinal;
    fi50_permissions : WORD;
    fi50_num_locks   : WORD;
    fi50_pathname    : PChar;
    fi50_username    : PChar;
    fi50_sharename   : PChar;
end;

```

```
type
```

```

TMibIfRow = packed record
    wszName           : array[0..255] of WideChar;
    dwIndex           : DWORD;
    dwType            : DWORD;
    dwMtu             : DWORD;
    dwSpeed           : DWORD;
    dwPhysAddrLen    : DWORD;
    bPhysAddr         : array[0..7] of Byte;
    dwAdminStatus     : DWORD;
    dwOperStatus     : DWORD;
    dwLastChange     : DWORD;
    dwInOctets        : DWORD;
    dwInUcastPkts    : DWORD;
    dwInNUCastPkts   : DWORD;
    dwInDiscards     : DWORD;
    dwInErrors        : DWORD;
    dwInUnknownProtos : DWORD;
    dwOutOctets       : DWORD;
    dwOutUcastPkts   : DWORD;
    dwOutNUCastPkts  : DWORD;
    dwOutDiscards    : DWORD;
    dwOutErrors       : DWORD;
    dwOutQLen        : DWORD;
    dwDescrLen        : DWORD;
    bDescr            : array[0..255] of Char;
end;
TMibIfArray = array [0..512] of TMibIfRow;
PMibIfRow = ^TMibIfRow;
PMibIfArray = ^TMibIfArray;

```

```
type
```

```

TMibIfTable = packed record
    dwNumEntries     : DWORD;
    Table            : TMibIfArray;
end;
PMibIfTable = ^TMibIfTable;

```

```

var
NetShareEnumNT:function (   servername:PWChar;
                           level:DWORD;
                           bufptr:Pointer;
                           prefmaxlen:DWORD;
                           entriesread,
                           totalentries,
                           resume_handle:LPDWORD): DWORD; stdcall;

var
NetShareEnum:function ( pszServer   : PChar;
                        sLevel      : Cardinal;
                        pbBuffer    : Pchar;
                        cbBuffer    : Cardinal;
                        pcEntriesRead,
                        pcTotalAvail: Pointer):DWORD; stdcall;

var
NetShareDelNT:function (servername: PWideChar;
                        netname: PWideChar;
                        reserved: DWORD): LongInt; stdcall;

var
NetShareDel:function (  pszServer,
                        pszNetName:PChar;
                        usReserved:Word): DWORD; stdcall;

var
NetShareAddNT: function(servername: PWideChar;
                        level: DWORD;
                        buf: Pointer;
                        parm_err: LPDWORD): DWORD; stdcall;

var
NetShareAdd: function ( pszServer:Pchar;
                        sLevel:Cardinal;
                        pbBuffer:PChar;
                        cbBuffer:Word):DWORD; stdcall;

Var
NetSessionEnumNT:function(servername,
                          UncClientName,
                          username:PWChar;
                          level:DWORD;
                          bufptr:Pointer;
                          prefmaxlen:DWORD;
                          entriesread,
                          totalentries,
                          resume_handle:LPDWORD):DWORD; stdcall;

var
NetSessionEnum:function(pszServer:PChar;
                        sLevel: DWORD;
                        pbBuffer:Pointer;
                        cbBuffer:DWORD;
                        pcEntriesRead,
                        pcTotalAvial:Pointer):integer; stdcall;

var
NetSessionDelNT:function(ServerName,
                          UncClientName,
                          username:PWChar):DWORD; stdcall;

var
NetSessionDel:function( pszServer:PChar;
                        pszClientName: PChar;

```

```

sReserved: SmallInt):DWORD; stdcall;

var
NetFileEnumNT:function( servername,
                        basepath,
                        username:PWChar;
                        level:DWORD;
                        bufptr:Pointer;
                        prefmaxlen:DWORD;
                        entriesread,
                        totalentries,
                        resume_handle:LPDWORD):DWORD; stdcall;

var
NetFileEnum:function(    pszServer,
                        pszBasePath:PChar;
                        sLevel:DWORD;
                        pbBuffer:Pointer;
                        cbBuffer:DWORD;
                        pcEntriesRead,
                        pcTotalAvail:pointer):integer; stdcall;

var
NetFileClose:function(  ServerName:PWideChar;
                        FileId:DWORD):DWORD; stdcall;

var
NetFileClose2:function( pszServer:PChar;
                        ulFileId:LongWord):DWORD; stdcall;

var
GetIfTable:function(    pIfTable      : PMibIfTable;
                        pdwSize       : PULONG;
                        bOrder        : Boolean ): DWORD; stdcall;

var
  MainForm: TMainForm;

implementation

{$R *.dfm}

{ TMainForm }

////////////////////////////////////
//
// Спочатку нам потрібно визначитися, під якою системою ми працюємо,
// щоб довідатися яку частину коду (для NT чи ні) використовувати в цей момент.
// Для цього напишемо невелику функцію, що і буде визначати тип системи.
//

function TMainForm.IsNT(var Value: Boolean): Boolean;
var Ver: TOSVersionInfo;
    BRes: Boolean;
begin
  Ver.dwOSVersionInfoSize := SizeOf(TOSVersionInfo);
  BRes := GetVersionEx(Ver);
  if not BRes then //Перевірка
  begin
    Result := False; //Інформація не отримана
    Exit;           //ідемо
  end else
    Result := True; //Інформація отримана

  case Ver.dwPlatformId of //визначаємося
    VER_PLATFORM_WIN32_NT      : Value := True; //Windows NT- підходить
    VER_PLATFORM_WIN32_WINDOWS : Value := False; //Windows 9 x-Me- підходить
    VER_PLATFORM_WIN32s       : Result := False //Windows 3.x- не підходить
  end;
end;

```

```

end;

////////////////////////////////////
//
// Одержання всіх відкритих загальних ресурсів
//

procedure TMainForm.btnGetSharesClick(Sender: TObject);
var
  i:Integer;
  FLibHandle : THandle;
  ShareNT : PShareInfo2Array; //<= Змінні
  entriesread,totalentries:DWORD; //<= для Windows NT
  Share : array [0..512] of TShareInfo50; //<= Змінні
  pcEntriesRead,pcTotalAvail:Word; //<= для Windows 9 x-Me
  OS: Boolean;
begin
  lbxShares.Items.Clear;
  if not IsNT(OS) then Close; //Визначаємо тип системи

  if OS then begin //Код для NT
    FLibHandle := LoadLibrary(' NETAPI32.DLL' ); //Завантажуємо бібліотеку
    if FLibHandle = 0 then Exit;
    //Зв' язуємо функцію
    @NetShareEnumNT := GetProcAddress(FLibHandle,' NetShareEnum' );
    if not Assigned(NetShareEnumNT) then //Перевірка
    begin
      FreeLibrary(FLibHandle);
      Exit;
    end;
    ShareNT := nil; //Очищаємо покажчик на масив структур
    //Виклик функції
    if NetShareEnumNT(nil,2,@ShareNT,DWORD(-1),
      @entriesread,@totalentries,nil) <> 0 then
    begin //Якщо виклик невдалий вивантажуємо бібліотеку
      FreeLibrary(FLibHandle);
      Exit;
    end;
    if entriesread > 0 then //Обробка результатів
    for i:= 0 to entriesread- 1 do
      lbxShares.Items.Add(String(ShareNT^[i].shi2_netname));
    end else begin //Код для 9 x-me
      FLibHandle := LoadLibrary(' SVRAPI.DLL' ); //Завантажуємо бібліотеку
      if FLibHandle = 0 then Exit;
      //Зв' язуємо функцію
      @NetShareEnum := GetProcAddress(FLibHandle,' NetShareEnum' );
      if not Assigned(NetShareEnum) then //Перевірка
      begin
        FreeLibrary(FLibHandle);
        Exit;
      end;
      //Виклик функції
      if NetShareEnum(nil,50,@Share,SizeOf(Share),
        @pcEntriesRead,@pcTotalAvail)<> 0 then
      begin //Якщо виклик невдалий вивантажуємо бібліотеку
        FreeLibrary(FLibHandle);
        Exit;
      end;
      if pcEntriesRead > 0 then //Обробка результатів
      for i:= 0 to pcEntriesRead- 1 do
        lbxShares.Items.Add(String(Share[i].shi50_netname));
      end;
      FreeLibrary(FLibHandle); //Не забуваємо вивантажити бібліотеку
    end;

    //////////////////////////////////////
    //
    // Закриття загального ресурсу
    //
  end;

```

```

procedure TMainForm.btnCloseSharesClick(Sender: TObject);
var
  OS:Boolean;
  FLibHandle : THandle;
  Name9x:array [0..12] of Char;
  NameNT:PWChar;
  i:Integer;
  ShareName: String;
begin
  if not IsNT(OS) then Close; //Визначаємо тип системи

  if lbxShares.Items.Count = 0 then Exit;
  for i:= 0 to lbxShares.Items.Count-1 do
    if lbxShares.Selected[i] then Break; //Шукаємо обраний елемент
  if not lbxShares.Selected[i] then Exit; //Якщо не знайдений ідемо
  ShareName := lbxShares.Items.Strings[i];

  if OS then begin //Код для NT
    FLibHandle := LoadLibrary(' NETAPI32.DLL' );
    if FLibHandle = 0 then Exit;
    @NetShareDelNT := GetProcAddress(FLibHandle,' NetShareDel' );
    if not Assigned(NetShareDelNT) then //Перевірка
    begin
      FreeLibrary(FLibHandle);
      Exit;
    end;
    i:= SizeOf(WideChar)*256;
    GetMem(NameNT,i); //Виділяємо пам' ять під змінну
    StringToWideChar(ShareName,NameNT,i); //Перетворимо в PWideChar
    NetShareDelNT(nil,NameNT,0); //Видаляємо ресурс
    FreeMem(NameNT); //Звільняємо пам' ять
  end else begin //Код для 9 x-ме
    FLibHandle := LoadLibrary(' SVRAPI.DLL' );
    if FLibHandle = 0 then Exit;
    @NetShareDel := GetProcAddress(FLibHandle,' NetShareDel' );
    if not Assigned(NetShareDel) then //Перевірка
    begin
      FreeLibrary(FLibHandle);
      Exit;
    end;
    FillChar(Name9x, SizeOf(Name9x), #0); //Очищаємо масив
    move(ShareName[1],Name9x[0],Length(ShareName)); //Заповнюємо масив
    NetShareDel(nil,@Name9x,0); //Видаляємо ресурс
  end;
  FreeLibrary(FLibHandle);
end;

////////////////////////////////////
//
// Показу діалогу вибору директорії
//

function TMainForm.SelectDirectory: String;
var
  lpItemID : PItemIDList;
  BrowseInfo : TBrowseInfo;
  DisplayName : array[0..MAX_PATH] of Char;
  TempPath : array[0..MAX_PATH] of Char;
begin
  FillChar(BrowseInfo, sizeof(TBrowseInfo), #0);
  BrowseInfo.hwndOwner := Handle;
  BrowseInfo.pszDisplayName := @DisplayName;
  BrowseInfo.lpszTitle := ' Specify a directory' ;
  BrowseInfo.ulFlags := BIF_RETURNONLYFSDIRS;
  lpItemID := SHBrowseForFolder(BrowseInfo);
  if Assigned(lpItemID) then begin
    SHGetPathFromIDList(lpItemID, TempPath);
    GlobalFreePtr(lpItemID);
  end;
end;

```

```

    end else Result := ' ';
    Result := String(TempPath);
end;

////////////////////////////////////
//
//  Додавання загального ресурсу
//

procedure TMainForm.btnAddSharesClick(Sender: TObject);
const
    STYPE_DISKTREE = 0;
    ACCESS_ALL = 258;
    SHI50F_FULL = 258;
var
    FLibHandle : THandle;
    Share9x : TShareInfo50;
    ShareNT : TShareInfo2;
    TmpDir, TmpName: String;
    TmpDirNT, TmpNameNT: PWChar;
    OS: Boolean;
    TmpLength: Integer;
begin
    TmpDir := SelectDirectory; //Визначаємо шлях до наступного ресурсу
    TmpName := InputBox(' Share name' , ' Enter name' , ' Test' ); //Визначаємо ім'
я під яким він буде видний у мережі
    if TmpDir = ' ' then Exit;

    if not IsNT(OS) then Close; //З' ясовуємо тип системи

    if OS then begin //Код для NT
        FLibHandle := LoadLibrary(' NETAPI32.DLL' );
        if FLibHandle = 0 then Exit;
        @NetShareAddNT := GetProcAddress(FLibHandle, ' NetShareAdd' );
        if not Assigned(NetShareAddNT) then
            begin
                FreeLibrary(FLibHandle);
                Exit;
            end;
        TmpLength := SizeOf(WideChar)*256; //Визначаємо необхідний розмір

        GetMem(TmpNameNT, TmpLength); //Конвертуємо в PWChar
        StringToWideChar(TmpName, TmpNameNT, TmpLength);
        ShareNT.shi2_netname := TmpNameNT; //Ім' я

        ShareNT.shi2_type := STYPE_DISKTREE; //Тип ресурсу
        ShareNT.shi2_remark := ' '; //Коментар
        ShareNT.shi2_permissions := ACCESS_ALL; //Доступ
        ShareNT.shi2_max_uses := DWORD(-1); // Кіл-У максим. підключ.
        ShareNT.shi2_current_uses := 0; // Кіл-У тік підкл.

        GetMem(TmpDirNT, TmpLength);
        StringToWideChar(TmpDir, TmpDirNT, TmpLength);
        ShareNT.shi2_path := TmpDirNT; //Шлях до ресурсу

        ShareNT.shi2_passwd := ' '; //Пароль

        NetShareAddNT(nil, 2, @ShareNT, nil); //Додаємо ресурс
        FreeMem (TmpNameNT); //звільняємо пам' ять
        FreeMem (TmpDirNT);
    end else begin
        FLibHandle := LoadLibrary(' SVRAPI.DLL' );
        if FLibHandle = 0 then Exit;
        @NetShareAdd := GetProcAddress(FLibHandle, ' NetShareAdd' );
        if not Assigned(NetShareAdd) then
            begin
                FreeLibrary(FLibHandle);
                Exit;
            end;
    end;
end;

```



```

begin
  lvSessions.Items.Clear;

  if not IsNT(OS) then Close; //З' ясовуємо тип системи

  if OS then begin //Код для NT
    FLibHandle := LoadLibrary(' NETAPI32.DLL' );
    if FLibHandle = 0 then Exit;
    @NetSessionEnumNT := GetProcAddress(FLibHandle, ' NetSessionEnum' );
    if not Assigned(NetSessionEnumNT) then
      begin
        FreeLibrary(FLibHandle);
        Exit;
      end;
    SessionInfo502 := nil;
    if NetSessionEnumNT(nil, nil, nil, 502, @SessionInfo502, DWORD(-
1), @entriesreadNT, @totalentries, nil)=0 then
      for i:=0 to EntriesReadNT-1 do
        begin
          with lvSessions.Items.Add do //Заповнення даними зі структури
            begin
              Caption := string(SessionInfo502^[i].sesi502_cname); //Ім' я комп' ютера
              SubItems.Add(SessionInfo502^[i].sesi502_username); //Ім' я користувача
              SubItems.Add(IntToStr(SessionInfo502^[i].sesi502_num_opens));
            //Відкритих ресурсів
              SubItems.Add(CardinalToTimeStr(SessionInfo502^[i].Sesi502_Time)); //Час
активний
              SubItems.Add(CardinalToTimeStr(SessionInfo502^[i].sesi502_idle_time));
            //Час не активний
              end;
            end;
          end else begin //Код для Windows 9 x-Me
            FLibHandle := LoadLibrary(' SVRAPI.DLL' );
            if FLibHandle = 0 then Exit;
            @NetSessionEnum := GetProcAddress(FLibHandle, ' NetSessionEnum' );
            if not Assigned(NetSessionEnum) then
              begin
                FreeLibrary(FLibHandle);
                Exit;
              end;
            if NetSessionEnum
(nil, 50, @SessionInfo50, SizeOf(SessionInfo50), @EntriesRead, @TotalAvial) = 0 then
              for i:=0 to EntriesRead-1 do
                begin
                  with lvSessions.Items.Add do //Заповнення даними зі структури
                    begin
                      Caption := string(SessionInfo50[i].Sesi50_cname); //Ім' я комп' ютера
                      SubItems.Add(SessionInfo50[i].Sesi50_username); //Ім' я користувача
                      SubItems.Add(IntToStr(SessionInfo50[i].sesi50_num_opens)); //Відкритих
ресурсів
                      SubItems.Add(CardinalToTimeStr(SessionInfo50[i].Sesi50_Time)); //Час
активний
                      SubItems.Add(CardinalToTimeStr(SessionInfo50[i].sesi50_idle_time));
                    //Час не активний
                      SessionCloseKey[i]:= SessionInfo50[i].sesi50_key; //Унікальний
ідентифікатор для закриття
                      end;
                    end;
                  end;
                FreeLibrary(FLibHandle);
                end;

                ////////////////////////////////////////////////////
                //
                // Завершення обраної сесії
                //

procedure TMainForm.btnCloseSessionClick(Sender: TObject);
var

```

```

OS: Boolean;
FLibHandle : THandle;
CNameNT: PWideChar;
CName9x: PAnsiChar;
Key:SmallInt;
i: Integer;
begin
  if not IsNT(OS) then Close; //3' ясовуємо тип системи

  if not Assigned(lvSessions.Selected) then Exit;
  i:= lvSessions.Selected.Index; //Визначаємо номер обраної сесії

  if OS then begin
    FLibHandle := LoadLibrary(' NETAPI32.DLL' );
    if FLibHandle = 0 then Exit;
    @NetSessionDelNT := GetProcAddress(FLibHandle, ' NetSessionDel' );
    if not Assigned(NetSessionDelNT) then
      begin
        FreeLibrary(FLibHandle);
        Exit;
      end;
    //Перетворимо дані в необхідний вид
    CNameNT := PWChar(WideString(' \\ ' +lvSessions.Items.Item[i].Caption));
    NetSessionDelNT(nil,CNameNT,nil);
  end else begin
    FLibHandle := LoadLibrary(' SVRAPI.DLL' );
    if FLibHandle = 0 then Exit;
    @NetSessionDel := GetProcAddress(FLibHandle, ' NetSessionDel' );
    if not Assigned(NetSessionDel) then
      begin
        FreeLibrary(FLibHandle);
        Exit;
      end;
    //Перетворимо дані в необхідний вид
    CName9x := PAnsiChar(lvSessions.Items.Item[i].Caption);
    key := SessionCloseKey[i]; //Беремо ключ із масиву
    NetSessionDel(nil,CName9x,Key);
  end;
  FreeLibrary(FLibHandle);
end;

////////////////////////////////////
//
// Одержання описку відкритих файлів
//

procedure TMainForm.btnGetFilesClick(Sender: TObject);
var
  OS: Boolean;
  FLibHandle : THandle;
  FileInfoNT: PFileInfo3Array;
  FileInfo9x: array [0..512] of TFileInfo50;
  TotalEntries,EntriesReadNT: DWORD;
  EntriesRead,TotalAvial: Word;
  i:integer;
begin
  lvfiles.Items.Clear;

  if not IsNT(OS) then Close; //3' ясовуємо тип системи

  if OS then begin //Код для NT
    FLibHandle := LoadLibrary(' NETAPI32.DLL' );
    if FLibHandle = 0 then Exit;
    @NetFileEnumNT := GetProcAddress(FLibHandle, ' NetFileEnum' );
    if not Assigned(NetFileEnumNT) then
      begin
        FreeLibrary(FLibHandle);
        Exit;
      end;

```

```

FileInfoNT := nil;
if NetFileEnumNT(nil,nil,nil,3,@FileInfoNT,DWORD(-1),@EntriesReadNT,
@totalentries, nil)=0 then
for i:=0 to EntriesReadNT-1 do
begin
with lvFiles.Items.Add do //Заповнення даними зі структури
begin
Caption := string(IntToStr(FileInfoNT^[i].fi3_id)); //Ідентифікатор
SubItems.Add(FileInfoNT^[i].fi3_pathname); //Шлях до файлу
SubItems.Add(FileInfoNT^[i].fi3_username); //Ім'я користувача
end;
end;
end else begin //Код для Windows 9 x-Me
FLibHandle := LoadLibrary(' SVRAPI.DLL' );
if FLibHandle = 0 then Exit;
@NetFileEnum := GetProcAddress(FLibHandle, ' NetFileEnum' );
if not Assigned(NetFileEnum) then
begin
FreeLibrary(FLibHandle);
Exit;
end;
if NetFileEnum (nil,
nil,50,@FileInfo9x,SizeOf(FileInfo9x),@EntriesRead,@TotalAvial)= 0 then
for i:=0 to EntriesRead-1 do
begin
with lvFiles.Items.Add do //Заповнення даними зі структури
begin
Caption := string(IntToStr(FileInfo9x[i].fi50_id)); //Ідентифікатор
SubItems.Add(FileInfo9x[i].fi50_pathname); //Шлях до файлу
SubItems.Add(FileInfo9x[i].fi50_username); //Ім'я користувача
end;
end;
end;
FreeLibrary(FLibHandle);
end;

////////////////////////////////////
//
// Закриття файлу
//

procedure TMainForm.btnCloseFileClick(Sender: TObject);
var
OS: Boolean;
FLibHandle : THandle;
i: Integer;
begin
if not IsNT(OS) then Close; //З'ясуємо тип системи

if not Assigned(lvFiles.Selected) then Exit;
i:= lvFiles.Selected.Index; //Визначаємо номер обраного файлу

if OS then begin //Код для NT
FLibHandle := LoadLibrary(' NETAPI32.DLL' );
if FLibHandle = 0 then Exit;
@NetFileClose := GetProcAddress(FLibHandle, ' NetFileClose' );
if not Assigned(NetFileClose) then
begin
FreeLibrary(FLibHandle);
Close;
end;
NetFileClose(nil,StrToInt(lvFiles.Items.Item[i].Caption)); //Закриваємо файл
end else begin //Код для Windows 9 x-Me
FLibHandle := LoadLibrary(' SVRAPI.DLL' );
if FLibHandle = 0 then Exit;
@NetFileClose2 := GetProcAddress(FLibHandle, ' NetFileClose2' );
if not Assigned(NetFileClose2) then
begin
FreeLibrary(FLibHandle);

```

```

        Close;
    end;
    NetFileClose2(nil, StrToInt(lvFiles.Items.Item[i].Caption));
end;
FreeLibrary(FLibHandle);
end;

////////////////////////////////////
//
//  Визначаємо вхідний- вихідний трафік
//

procedure TMainForm.tmrTrafficTimer(Sender: TObject);
    // Допоміжна функція, що перетворить MAC адресу до "нормального" виду
    // Визначаємо спеціальний тип, щоб можна було передати у функцію масив
    type TMAC = array [0..7] of Byte;
    // Як перше значення масив, друге значення, розмір даних у масиві
    function GetMAC(Value: TMAC; Length: DWORD): String;
    var
        i: Integer;
    begin
        if Length = 0 then Result := ' 00-00-00' else
        begin
            Result := ' ';
            for i:= 0 to Length-2 do
                Result := Result + IntToHex(Value[i],2)+' -';
            Result := Result + IntToHex(Value[ Length-1],2);
        end;
    end;

//Сама процедура
var
    FLibHandle : THandle;
    Table: TMibIfTable;
    i : integer;
    Size : integer;
begin
    tmrTraffic.Enabled := false; //Припиняємо про всякий випадок таймер
    lvTraffic.Items.BeginUpdate;
    lvTraffic.Items.Clear; //Очищаємо список
    FLibHandle := LoadLibrary(' IPHLPAPI.DLL' ); //Завантажуємо бібліотеку
    if FLibHandle = 0 then Exit;
    @GetIfTable := GetProcAddress(FLibHandle, ' GetIfTable' );
    if not Assigned(GetIfTable) then
    begin
        FreeLibrary(FLibHandle);
        Close;
    end;

    Size := SizeOf(Table);
    if GetIfTable(@Table, @Size, false ) = 0 then //Виконуємо функцію
        for i:= 0 to Table.dwNumEntries-1 do begin
            with lvTraffic.Items.Add do begin //Виводимо результати
                Caption := String(Table.Table[i].bDescr); //Найменування інтерфейсу
                SubItems.Add(GetMAC(TMAC(Table.Table[i].bPhysAddr),
                    Table.Table[i].dwPhysAddrLen)); //MAC адреса
                SubItems.Add(IntToStr(Table.Table[i].dwInOctets)); //Усього прийнято
байт
                SubItems.Add(IntToStr(Table.Table[i].dwOutOctets)); //Усього відправлено
байт
            end;
        end;
    lvTraffic.Items.EndUpdate;
    FreeLibrary(FLibHandle);
    tmrTraffic.Enabled := true; //Не забуваємо активувати таймер
end;

```

```

function OpenEnum(const NetContainerToOpen: PNetResource; ResScope, ResType,
ResUsage: DWORD): THandle;
var
  hNetEnum: THandle;
begin
  Result:=0;
  if (NO_ERROR<>WNetOpenEnum(ResScope, ResType, ResUsage,
                             NetContainerToOpen, hNetEnum))
  then ShowMessage( ' Помилка!' )
  else Result:=hNetEnum;
end;

function EnumResources(const ParentNode: TTreeNode;
ResScope, ResType, ResUsage: DWORD; hNetEnum: THandle): UINT;
function ShowResource(const ParentNode: TTreeNode; Res: TNetResource):
TTreeNode;
begin
  Result:=MainForm.NetTree.Items.AddChild(ParentNode,
string(Res.lpRemoteName));
end;

const
  RESOURCE_BUF_ENTRIES = 2000;

var
  ResourceBuffer: array[1..RESOURCE_BUF_ENTRIES] of TNetResource;
  i, ResourceBuf, EntriesToGet: dword;
  NewNode: TTreeNode;
begin
  Result:=0;
  while true do
  begin
    ResourceBuf:=sizeof(ResourceBuffer);
    EntriesToGet:=RESOURCE_BUF_ENTRIES;
    if (NO_ERROR<>WNetEnumResource(hNetEnum, EntriesToGet,
                                   @ResourceBuffer, ResourceBuf))
    then
      begin
        case GetLastError() of
          NO_ERROR: // проход буферу без перемикання
            Break;
          ERROR_NO_MORE_ITEMS:
            // Повертає о у тому випадку, коли останов
            // RESOURCE_BUF_ENTRIES данні на попередньому виклику, щоб
            // WNetEnumResource, та були точно
            // RESOURCE_BUF_ENTRIES данні в запису на момент
            // попереднього виклику
            Exit;
          else ShowMessage(Помилка!' );
            Result:=1;
            Exit;
        end;
      end;
    for i:=1 to EntriesToGet do
      begin
        NewNode:=ShowResource(ParentNode, ResourceBuffer[i]);
        if (ResourceBuffer[i].dwUsage and RESOURCEUSAGE_CONTAINER)<>0
        then MainForm.Open_Do_Close_Enum(NewNode, ResScope, ResType, ResUsage,
@ResourceBuffer[i]);
        Application.ProcessMessages;
      end;
    end;
  end;

procedure TMainForm.Open_Do_Close_Enum(const ParentNode: TTreeNode; ResScope,
ResType, ResUsage: DWORD; const NetContainerToOpen: PNetResource);
var
  hNetEnum: THandle;
begin

```

```

hNetEnum:=OpenEnum(NetContainerToOpen, ResScope, ResType, ResUsage);
if (hNetEnum=0)
then Exit;
EnumResources(ParentNode, ResScope, ResType, ResUsage, hNetEnum);
if (NO_ERROR<>WNetCloseEnum(hNetEnum))
then ShowMessage(' WNetCloseEnum Помилка' );
end;

procedure TMainForm.Button1Click(Sender: TObject);
var
  ResScope, ResType, ResUsage: dword;
begin
  Button1.Caption:=' Пошук мережних ресурсів. Чекайте...' ;
  Button1.Enabled:=false;
  //
  NetTree.Items.Clear;
  case rgScope.ItemIndex of
    1: ResScope:=RESOURCE_GLOBALNET;
    2: ResScope:=RESOURCE_REMEMBERED;
    else ResScope:=RESOURCE_CONNECTED;
  end;
  ResType:=0;
  if cbTypeAny.Checked
  then ResType:=ResType or RESOURCETYPE_ANY;
  if cbTypeDisk.Checked
  then ResType:=ResType or RESOURCETYPE_DISK;
  if cbTypePrint.Checked
  then ResType:=ResType or RESOURCETYPE_PRINT;
  ResUsage:=0;
  if cbUsageConnectable.Checked
  then ResUsage:=ResUsage or RESOURCEUSAGE_CONNECTABLE;
  if cbUsageContainer.Checked
  then ResUsage:=ResUsage or RESOURCEUSAGE_CONTAINER;
  Open_Do_Close_Enum(NetTree.Items.Add(nil, ' Network Resources' ),
    ResScope, ResType, ResUsage, nil);
  //
  Button1.Caption:=' Обновити список ресурсів' ;
  Button1.Enabled:=true;

end;

procedure TMainForm.NetTreeCustomDrawItem(Sender: TCustomTreeView;
  Node: TTreeNode; State: TCustomDrawState; var DefaultDraw: Boolean);
begin
  if cdsSelected in State
  then Sender.Canvas.Font.Style:=Sender.Canvas.Font.Style+[fsUnderline];
end;

procedure TMainForm.NetTreeDbClick(Sender: TObject);
begin
  ShellExecute(0, ' open' , PChar(NetTree.Selected.Text), ' \ ' , ' \ ' , SW_SHOW);
end;

procedure TMainForm.NetTreeGetImageIndex(Sender: TObject; Node: TTreeNode);
begin
  if Node.HasChildren
  then Node.ImageIndex:=1
  else Node.ImageIndex:=0;
end;

procedure TMainForm.Button4Click(Sender: TObject);
begin
  Form1.Show;
end;

procedure TMainForm.Button2Click(Sender: TObject);
begin
  Form2.Show;
end;

```

```
procedure TMainForm.Button3Click(Sender: TObject);  
begin  
  Form3.Show;  
end;  
  
end.
```

К6П3_2023

Файл IPHLPAPI.pas- обробка API функцій

```

unit IPHLPAPI;

interface
uses
  Windows, winsock;

const
  VERSION = ' 1.5' ;

//----- Заголовок з Microsoft IPTYPES.H-----

const
  ANY_SIZE = 1;
  MAX_ADAPTER_DESCRIPTION_LENGTH = 128; // arb.
  MAX_ADAPTER_NAME_LENGTH = 256; // змінна
  MAX_ADAPTER_ADDRESS_LENGTH = 8; // змінна
  DEFAULT_MINIMUM_ENTITIES = 32; // змінна
  MAX_HOSTNAME_LEN = 128; // змінна
  MAX_DOMAIN_NAME_LEN = 128; // змінна
  MAX_SCOPE_ID_LEN = 256; // змінна

  // Вузлові типи ( NETBIOS)
  BROADCAST_NODETYPE = 1;
  PEER_TO_PEER_NODETYPE = 2;
  MIXED_NODETYPE = 4;
  HYBRID_NODETYPE = 8;

  NETBIOSTypes : array[0..8] of string[20] =
    ( ' Невизначений' , ' Передача' , ' Рівень до рівня' , ' ' , '
Змішаний' , ' ' , ' ' , ' ' , ' Гібрид'
    );

  // Типи адаптеру
  { v1.4-> 1.5
  IF_OTHER_ADAPTERTYPE = 0;
  IF_ETHERNET_ADAPTERTYPE = 1;
  IF_TOKEN_RING_ADAPTERTYPE = 2;
  IF_FDDI_ADAPTERTYPE = 3;
  IF_PPP_ADAPTERTYPE = 4;
  IF_LOOPBACK_ADAPTERTYPE = 5;
  IF_SLIP_ADAPTERTYPE = 6;

  found in ipifcons.h :
  #define MIB_IF_TYPE_OTHER 1
  #define MIB_IF_TYPE_ETHERNET 6
  #define MIB_IF_TYPE_TOKENRING 9
  #define MIB_IF_TYPE_FDDI 15
  #define MIB_IF_TYPE_PPP 23
  #define MIB_IF_TYPE_LOOPBACK 24
  #define MIB_IF_TYPE_SLIP 28
  }
  IF_OTHER_ADAPTERTYPE = 1;
  IF_ETHERNET_ADAPTERTYPE = 6;
  IF_TOKEN_RING_ADAPTERTYPE = 9;
  IF_FDDI_ADAPTERTYPE = 15;
  IF_PPP_ADAPTERTYPE = 23;
  IF_LOOPBACK_ADAPTERTYPE = 24;
  IF_SLIP_ADAPTERTYPE = 28;

  // AdaptTypes : array[0..6] of string[10] =
  // ( ' інший' , ' ethernet' , ' tokenring' , ' FDDI' , ' PPP' , '
loopback' , ' SLIP' );
  AdaptTypes : array[1..28] of string[10] =

```

```

( 'інший' , ' ' , ' ' , ' ' , ' ' , ' ' , ' ethernet' , ' ' , ' ' , '
tokenring' , ' ' , ' ' , ' ' , ' ' , ' ' , ' FDDI' , ' ' , ' ' , ' ' , ' ' ,
' ' , ' ' , ' PPP' ,
' loopback' , ' ' , ' ' , ' ' , ' SLIP' );
// Кінець змін в типі адаптерів

//-----для інших MS заготовочних файлів-----

MAX_INTERFACE_NAME_LEN = 256; { mrap1.h }
MAXLEN_PHYSADDR = 8; { iprtmib.h }
MAXLEN_IFDESCR = 256; {"-- }

//-----

type
  TMacAddress = array[1..MAX_ADAPTER_ADDRESS_LENGTH] of byte;

//---IP адресні структури-----

PTIP_ADDRESS_STRING = ^TIP_ADDRESS_STRING;
TIP_ADDRESS_STRING = array[0..15] of char; // IP рядок
//
PTIP_ADDR_STRING = ^TIP_ADDR_STRING;
TIP_ADDR_STRING = packed record // для використання у зв'язних списках
  Next: PTIP_ADDR_STRING;
  IpAddress: TIP_ADDRESS_STRING;
  IpMask: TIP_ADDRESS_STRING;
  Context: DWORD;
end;

//-----Fixed Info структура-----

PTFixedInfo = ^TFixedInfo;
TFixedInfo = packed record
  HostName: array[1..MAX_HOSTNAME_LEN + 4] of char; // данні
  DomainName: array[1..MAX_DOMAIN_NAME_LEN + 4] of char; // данні
  CurrentDNSServer: PTIP_ADDR_STRING;
  DNSServerList: TIP_ADDR_STRING;
  NodeType: UINT;
  ScopeID: array[1..MAX_SCOPE_ID_LEN + 4] of char; // данні
  EnableRouting: UINT;
  EnableProxy: UINT;
  EnableDNS: UINT;
end;

//-----структура мережного інтерфейсу-----

////////////////////////////////////
//
//
// Наступне є діючими станами для WAN да LAN інтерфейсів. //
// Порядок станів створений для визначення. Для //
// стану >= CONNECTED можливо передавати данні зразу. Стан >= DISCONNECTED
//
// може передавати деякі данні. Стан < DISCONNECTED може //
// не передавати дані.
//
// карта з поміткою UNREACHABLE якщо DIM викликає InterfaceUnreachable для
//
// причин. Крім невдачі з'єднання. //
//
//
// NON_OPERATIONAL- Перевірка для LAN інтерфейсу. Позначає карту що не
працює //
// або не з'єднується з картою. //
// UNREACHABLE- Перевірка WAN інтерфейсів . Позначає, що віддалений сайт
//

```

```

//                                     не з'єднується за потрібний час.
//
// DISCONNECTED- Перевірка WAN інтерфейсів . Позначає, що віддалений сайт
//
//                                     не з'єднується.
//
// CONNECTING- Перевірка WAN інтерфейсів . Означає спробу з'єднання //
//                                     з сайтом, якого немає. //
// CONNECTED- Перевірка WAN інтерфейсів . Позначає, що віддалений сайт
//
//                                     з'єднується.
//
// OPERATIONAL- Перевірка LAN Interfaces. Позначає карту підключену //
//                                     в праці. //
//
//
// Усі дії користувачів записуються до MIB-II значення //
// можуть бути використовані //
//
//
////////////////////////////////////

const
// данні додані до ipifcons.h
IF_OPER_STATUS_NON_OPERATIONAL = 0 ;
IF_OPER_STATUS_UNREACHABLE = 1 ;
IF_OPER_STATUS_DISCONNECTED = 2 ;
IF_OPER_STATUS_CONNECTING = 3 ;
IF_OPER_STATUS_CONNECTED = 4 ;
IF_OPER_STATUS_OPERATIONAL = 5 ;

MIB_IF_TYPE_OTHER = 1 ;
MIB_IF_TYPE_ETHERNET = 6 ;
MIB_IF_TYPE_TOKENRING = 9 ;
MIB_IF_TYPE_FDDI = 15 ;
MIB_IF_TYPE_PPP = 23 ;
MIB_IF_TYPE_LOOPBACK = 24 ;
MIB_IF_TYPE_SLIP = 28 ;

MIB_IF_ADMIN_STATUS_UP = 1 ;
MIB_IF_ADMIN_STATUS_DOWN = 2 ;
MIB_IF_ADMIN_STATUS_TESTING = 3 ;

MIB_IF_OPER_STATUS_NON_OPERATIONAL = 0 ;
MIB_IF_OPER_STATUS_UNREACHABLE = 1 ;
MIB_IF_OPER_STATUS_DISCONNECTED = 2 ;
MIB_IF_OPER_STATUS_CONNECTING = 3 ;
MIB_IF_OPER_STATUS_CONNECTED = 4 ;
MIB_IF_OPER_STATUS_OPERATIONAL = 5 ;

type
PTMibIfRow = ^TMibIfRow;
TMibIfRow = packed record
    wszName: array[1..MAX_INTERFACE_NAME_LEN] of WCHAR;
    dwIndex: DWORD;
    dwType: DWORD; // дивись MIB_IF_TYPE
    dwMTU: DWORD;
    dwSpeed: DWORD;
    dwPhysAddrLen: DWORD;
    bPhysAddr: array[1..MAXLEN_PHYSADDR] of byte;
    dwAdminStatus: DWORD; // дивись MIB_IF_ADMIN_STATUS
    dwOperStatus: DWORD; // дивись MIB_IF_OPER_STATUS
    dwLastChange: DWORD;
    dwInOctets: DWORD;
    dwInUcastPkts: DWORD;
    dwInNUCcastPkts: DWORD;
    dwInDiscards: DWORD;
    dwInErrors: DWORD;
    dwInUnknownProtos: DWORD;

```

```

    dwOutOctets: DWORD;
    dwOutUCastPkts: DWORD;
    dwOutNUCastPkts: DWORD;
    dwOutDiscards: DWORD;
    dwOutErrors: DWORD;
    dwOutQLen: DWORD;
    dwDescrLen: DWORD;
    bDescr: array[1..MAXLEN_IFDESCR] of char; //byte;
end;

//
PTMibIfTable = ^TMibIfTable;
TMibIfTable = packed record
    dwNumEntries: DWORD;
    Table: array[0..ANY_SIZE- 1] of TMibIfRow;
end;

//---ADAPTER INFO структура-----

PTIP_ADAPTER_INFO = ^TIP_ADAPTER_INFO;
TIP_ADAPTER_INFO = packed record
    Next: PTIP_ADAPTER_INFO;
    ComboIndex: DWORD;
    AdapterName: array[1..MAX_ADAPTER_NAME_LENGTH + 4] of char; //
данні
    Description: array[1..MAX_ADAPTER_DESCRIPTION_LENGTH + 4] of char;
// данні
    AddressLength: UINT;
    Address: array[1..MAX_ADAPTER_ADDRESS_LENGTH] of byte; // данні
    Index: DWORD;
    aType: UINT;
    DHCPEnabled: UINT;
    CurrentIPAddress: TIP_ADDR_STRING;
    IPAddressList: TIP_ADDR_STRING;
    GatewayList: TIP_ADDR_STRING;
    DHCPServer: TIP_ADDR_STRING;
    HaveWINS: BOOL;
    PrimaryWINSServer: TIP_ADDR_STRING;
    SecondaryWINSServer: TIP_ADDR_STRING;
    LeaseObtained: LongInt ; // UNIX час, секунди з 1970
    LeaseExpires: LongInt; // UNIX час, секунди з 1970
    SpareStuff: array [1..200] of char ; // данні- простір для списку IP
адрес
end;

//-----TCP структура-----

PTMibTCPRow = ^TMibTCPRow;
TMibTCPRow = packed record
    dwState: DWORD;
    dwLocalAddr: DWORD;
    dwLocalPort: DWORD;
    dwRemoteAddr: DWORD;
    dwRemotePort: DWORD;
end;
//
PTMibTCPTable = ^TMibTCPTable;
TMibTCPTable = packed record
    dwNumEntries: DWORD;
    Table: array[0..0] of TMibTCPRow;
end;
//
PTMibTCPStats = ^TMibTCPStats;
TMibTCPStats = packed record
    dwRTOAlgorithm: DWORD;
    dwRTOMin: DWORD;
    dwRTOMax: DWORD;
    dwMaxConn: DWORD;
    dwActiveOpens: DWORD;

```

```

    dwPassiveOpens: DWORD;
    dwAttemptFails: DWORD;
    dwEstabResets: DWORD;
    dwCurrEstab: DWORD;
    dwInSegs: DWORD;
    dwOutSegs: DWORD;
    dwRetransSegs: DWORD;
    dwInErrs: DWORD;
    dwOutRsts: DWORD;
    dwNumConns: DWORD;
end;

```

```
//-----UDP CTPYKTYPA -----
```

```

PTMibUDPRow = ^TMibUDPRow;
TMibUDPRow = packed record
    dwLocalAddr: DWORD;
    dwLocalPort: DWORD;
end;
//
PTMibUDPTable = ^TMIBUDPTable;
TMIBUDPTable = packed record
    dwNumEntries: DWORD;
    UDPTable: array[0..ANY_SIZE- 1] of TMibUDPRow;
end;
//
PTMibUdpStats = ^TMIBUdpStats;
TMIBUdpStats = packed record
    dwInDatagrams: DWORD;
    dwNoPorts: DWORD;
    dwInErrors: DWORD;
    dwOutDatagrams: DWORD;
    dwNumAddrs: DWORD;
end;

```

```
//-----IP CTPYKTYPA -----
```

```

//
PTMibIPNetRow = ^TMibIPNetRow;
TMibIPNetRow = packed record
    dwIndex: DWord;
    dwPhysAddrLen: DWord;
    bPhysAddr: TMacAddress;
    dwAddr: DWord;
    dwType: DWord;
end;
//
PTMibIPNetTable = ^TMibIPNetTable;
TMibIPNetTable = packed record
    dwNumEntries: DWORD;
    Table: array[0..ANY_SIZE- 1] of TMibIPNetRow;
end;
//
PTMibIPStats = ^TMibIPStats;
TMibIPStats = packed record
    dwForwarding: DWORD;
    dwDefaultTTL: DWORD;
    dwInReceives: DWORD;
    dwInHdrErrors: DWORD;
    dwInAddrErrors: DWORD;
    dwForwDatagrams: DWORD;
    dwInUnknownProtos: DWORD;
    dwInDiscards: DWORD;
    dwInDelivers: DWORD;
    dwOutRequests: DWORD;
    dwRoutingDiscards: DWORD;
    dwOutDiscards: DWORD;
    dwOutNoRoutes: DWORD;
    dwReasmTimeOut: DWORD;

```

```

    dwReasmReqds: DWORD;
    dwReasmOKs: DWORD;
    dwReasmFails: DWORD;
    dwFragOKs: DWORD;
    dwFragFails: DWORD;
    dwFragCreates: DWORD;
    dwNumIf: DWORD;
    dwNumAddr: DWORD;
    dwNumRoutes: DWORD;
end;
//
PTMibIPAddrRow = ^TMibIPAddrRow;
TMibIPAddrRow = packed record
    dwAddr: DWORD;
    dwIndex: DWORD;
    dwMask: DWORD;
    dwBCastAddr: DWORD;
    dwReasmSize: DWORD;
    Unused1,
    Unused2: WORD;
end;
//
PTMibIPAddrTable = ^TMibIPAddrTable;
TMibIPAddrTable = packed record
    dwNumEntries: DWORD;
    Table: array[0..ANY_SIZE- 1] of TMibIPAddrRow;
end;

//
PTMibIPForwardRow = ^TMibIPForwardRow;
TMibIPForwardRow = packed record
    dwForwardDest: DWORD;
    dwForwardMask: DWORD;
    dwForwardPolicy: DWORD;
    dwForwardNextHop: DWORD;
    dwForwardIFIndex: DWORD;
    dwForwardType: DWORD;
    dwForwardProto: DWORD;
    dwForwardAge: DWORD;
    dwForwardNextHopAS: DWORD;
    dwForwardMetric1: DWORD;
    dwForwardMetric2: DWORD;
    dwForwardMetric3: DWORD;
    dwForwardMetric4: DWORD;
    dwForwardMetric5: DWORD;
end;
//
PTMibIPForwardTable = ^TMibIPForwardTable;
TMibIPForwardTable = packed record
    dwNumEntries: DWORD;
    Table: array[0..ANY_SIZE- 1] of TMibIPForwardRow;
end;

//----- ICMP-CTPVKTYPA -----

PTMibICMPStats = ^TMibICMPStats;
TMibICMPStats = packed record
    dwMsgs: DWORD;
    dwErrors: DWORD;
    dwDestUnreaches: DWORD;
    dwTimeEcxcds: DWORD;
    dwParmProbs: DWORD;
    dwSrcQuenches: DWORD;
    dwRedirects: DWORD;
    dwEchos: DWORD;
    dwEchoReps: DWORD;
    dwTimeStamps: DWORD;
    dwTimeStampReps: DWORD;
    dwAddrMasks: DWORD;

```

```

    dwAddrReps: DWORD;
end;

PTMibICMPInfo = ^TMibICMPInfo;
TMibICMPInfo = packed record
    InStats: TMibICMPStats;
    OutStats: TMibICMPStats;
end;

//-----импорт до IPHLPAPI.DLL-----

var

GetAdaptersInfo: function ( pAdapterInfo: PTIP_ADAPTER_INFO;
    pOutBufLen: PULONG ): DWORD; stdcall;

GetNetworkParams: function ( FixedInfo: PTFixedInfo; pOutPutLen: PULONG ):
    DWORD; stdcall;

GetTcpTable: function ( pTCPTable: PTMibTCPTable; pDwSize: PDWORD;
    bOrder: BOOL ): DWORD; stdcall;

GetTcpStatistics: function ( pStats: PTMibTCPStats ): DWORD; stdcall;

GetUdpTable: function ( pUdpTable: PTMibUDPTable; pDwSize: PDWORD;
    bOrder: BOOL ): DWORD; stdcall;

GetUdpStatistics: function ( pStats: PTMibUdpStats ): DWORD; stdcall;

GetIpStatistics: function ( pStats: PTMibIPStats ): DWORD; stdcall;

GetIpNetTable: function ( pIpNetTable: PTMibIPNetTable;
    pdwSize: PULONG; bOrder: BOOL ): DWORD; stdcall;

GetIpAddrTable: function ( pIpAddrTable: PTMibIPAddrTable;
    pdwSize: PULONG; bOrder: BOOL ): DWORD; stdcall;

GetIpForwardTable: function ( pIPForwardTable: PTMibIPForwardTable;
    pdwSize: PULONG; bOrder: BOOL ): DWORD; stdCall;

GetIcmpStatistics: function ( pStats: PTMibICMPInfo ): DWORD; stdCall;

GetRTTAndHopCount: function ( DestIPAddress: DWORD; HopCount: PULONG;
    MaxHops: ULONG; RTT: PULONG ): BOOL; stdCall;

GetIfTable: function ( pIfTable: PTMibIfTable; pdwSize: PULONG;
    bOrder: boolean ): DWORD; stdCall;

GetIfEntry: function ( pIfRow: PTMibIfRow ): DWORD; stdCall;

// попередження - недокументована функція, можливі баги при використанні
GetFriendlyIfIndex: function (var IfIndex: DWORD): DWORD; stdcall;

const
    IpHlpDLL = 'IPHLPAPI.DLL' ;
var
    IpHlpModule: THandle;

    function LoadIpHlp: Boolean;

implementation

function LoadIpHlp: Boolean;
begin
    Result := True;
    if IpHlpModule <> 0 then Exit;

// відкрити DLL

```

```

IpHlpModule := LoadLibrary (IpHlpDLL);
if IpHlpModule = 0 then
begin
    Result := false;
    exit ;
end ;
GetAdaptersInfo := GetProcAddress (IpHlpModule, ' GetAdaptersInfo' ) ;
GetNetworkParams := GetProcAddress (IpHlpModule, ' GetNetworkParams' )
;

GetTcpTable := GetProcAddress (IpHlpModule, ' GetTcpTable' ) ;
GetTcpStatistics := GetProcAddress (IpHlpModule, ' GetTcpStatistics' )
;

GetUdpTable := GetProcAddress (IpHlpModule, ' GetUdpTable' ) ;
GetUdpStatistics := GetProcAddress (IpHlpModule, ' GetUdpStatistics' )
;

GetIpStatistics := GetProcAddress (IpHlpModule, ' GetIpStatistics' ) ;
GetIpNetTable := GetProcAddress (IpHlpModule, ' GetIpNetTable' ) ;
GetIpAddrTable := GetProcAddress (IpHlpModule, ' GetIpAddrTable' ) ;
GetIpForwardTable := GetProcAddress (IpHlpModule, ' GetIpForwardTable'
) ;
GetIcmpStatistics := GetProcAddress (IpHlpModule, ' GetIcmpStatistics'
) ;
GetRTTAndHopCount := GetProcAddress (IpHlpModule, ' GetRTTAndHopCount'
) ;

GetIfTable := GetProcAddress (IpHlpModule, ' GetIfTable' ) ;
GetIfEntry := GetProcAddress (IpHlpModule, ' GetIfEntry' ) ;
GetFriendlyIfIndex := GetProcAddress (IpHlpModule, '
GetFriendlyIfIndex' ) ;
end;

initialization
    IpHlpModule := 0 ;
finalization
    if IpHlpModule <> 0 then
    begin
        FreeLibrary (IpHlpModule) ;
        IpHlpModule := 0 ;
    end ;

end.

```

Файл IPHelper.pas - функції роботи з IP-протоколом

```

unit IPHelper;

interface

uses
  Windows, Messages, SysUtils, Classes, Dialogs, IpHlpApi;

const
  NULL_IP      = ' 0.0. 0.0' ;

//---перетворення добре відомих номерів портів до імен сервісів-----

type
  TWellKnownPort = record
    Prt: DWORD;
    Srv: string[20];
  end;

const
  // тільки найбільш популярні сервіси...
  WellKnownPorts: array[1..32] of TWellKnownPort
  = (
//    ( Prt: 0; Srv:  ' RESRVED' ),      {Зарезервовано}
    ( Prt: 7; Srv:  ' ECHO  ' ),      {Ping      }
    ( Prt: 9; Srv:  ' DISCARD' ),
    ( Prt: 13; Srv: ' DAYTIME' ),
    ( Prt: 17; Srv: ' QOTD  ' ),      {Показчик на день}
    ( Prt: 19; Srv: ' CHARGEN' ),     {Генератор символів}
    ( Prt: 20; Srv: ' FTPDATA' ),     { File Transfer Protocol- данні}
    ( Prt: 21; Srv: ' FTPCTRL' ),     { File Transfer Protocol- управління}
    ( Prt: 22; Srv: ' SSH   ' ),
    ( Prt: 23; Srv: ' TELNET ' ),
    ( Prt: 25; Srv: ' SMTP  ' ),      { Simple Mail Transfer Protocol}
    ( Prt: 37; Srv: ' TIME  ' ),      { Часовий протокол }
    ( Prt: 43; Srv: ' WHOIS ' ),      { Сервіс - Кто це }
    ( Prt: 53; Srv: ' DNS   ' ),      { Domain Name Service }
    ( Prt: 67; Srv: ' BOOTPS ' ),     { BOOTP Сервер }
    ( Prt: 68; Srv: ' BOOTPC ' ),     { BOOTP Кієнт }
    ( Prt: 69; Srv: ' TFTP  ' ),      { стандартний FTP }
    ( Prt: 70; Srv: ' GOPHER ' ),     { Протокол Gopher }
    ( Prt: 79; Srv: ' FINGER ' ),     { Протокол Finger }
    ( Prt: 80; Srv: ' HTTP  ' ),      { Протокол HTTP }
    ( Prt: 88; Srv: ' KERBROS' ),     { Протокол Kerberos }
    ( Prt: 109; Srv: ' POP2  ' ),     { Протокол Post Office Protocol Version
2 }
    ( Prt: 110; Srv: ' POP3  ' ),     { Протокол Post Office Protocol Version
3 }
    ( Prt: 111; Srv: ' SUN_RPC' ),     { Протокол SUN Remote Procedure Call }
    ( Prt: 119; Srv: ' NNTP  ' ),     { Протокол Network News Transfer
Protocol }
    ( Prt: 123; Srv: ' NTP   ' ),     { Протокол Network Time protocol
}
    ( Prt: 135; Srv: ' DCOMRPC' ),     { Протокол Location Service
}
    ( Prt: 137; Srv: ' NBNAME ' ),     { NETBIOS сервіс імен }
    ( Prt: 138; Srv: ' NBDGRAM' ),     { NETBIOS сервіс датаграм }
    ( Prt: 139; Srv: ' NBSESS ' ),     { NETBIOS сервіс сесій }
    ( Prt: 143; Srv: ' IMAP  ' ),     { Протокол Internet Message Access
Protocol }
    ( Prt: 161; Srv: ' SNMP  ' ),     { Протокол Simple Netw. Management
Protocol }
    ( Prt: 169; Srv: ' SEND  ' )
  )

```

```

);

//-----перетворення ICMP кодів помилок до рядків-----

const
  ICMP_ERROR_BASE = 11000;
  IcmpErr : array[1..22] of string =
  (
    ' IP_BUFFER_TOO_SMALL' , ' IP_DEST_NET_UNREACHABLE' , '
IP_DEST_HOST_UNREACHABLE' ,
    ' IP_PROTOCOL_UNREACHABLE' , ' IP_DEST_PORT_UNREACHABLE' , ' IP_NO_RESOURCES'
  ,
    ' IP_BAD_OPTION' , ' IP_HARDWARE_ПОМИЛКА' , ' IP_PACKET_TOO_BIG' , '
IP_REQUEST_TIMED_OUT' ,
    ' IP_BAD_REQUEST' , ' IP_BAD_ROUTE' , ' IP_TTL_EXPIRED_TRANSIT' ,
    ' IP_TTL_EXPIRED_REASSEM' , ' IP_PARAMETER_PROBLEM' , ' IP_SOURCE_QUENCH' ,
    ' IP_OPTION_TOO_BIG' , ' IP_BAD_DESTINATION' , ' IP_ADDRESS_DELETED' ,
    ' IP_SPEC_MTU_CHANGE' , ' IP_MTU_CHANGE' , ' IP_UNLOAD'
  );

//-----перетворення різних перерахованих величин у рядки-----

ARPEntryType : array[1..4] of string = ( ' інший' , ' неправильний' ,
  ' динамічний' , ' статичний'
);
TCPConnState :
  array[1..12] of string =
  ( ' closed' , ' listening' , ' syn_sent' ,
    ' syn_rcvd' , ' established' , ' fin_wait1' ,
    ' fin_wait2' , ' close_wait' , ' closing' ,
    ' last_ack' , ' time_wait' , ' delete_tcb'
  );
TCPToAlgo : array[1..4] of string =
  ( ' Const.Timeout' , ' MIL-STD-1778' ,
    ' Van Jacobson' , ' інший' );
IPForwTypes : array[1..4] of string =
  ( ' інший' , ' invalid' , ' local' , ' remote' );
IPForwProtos : array[1..18] of string =
  ( ' інший' , ' LOCAL' , ' NETMGMT' , ' ICMP' , ' EGP' ,
    ' GGP' , ' HELLO' , ' RIP' , ' IS_IS' , ' ES_IS' ,
    ' CISCO' , ' BBN' , ' OSPF' , ' BGP' , ' BOOTP' ,
    ' AUTO_STAT' , ' STATIC' , ' NOT_DOD' );

type
// для IpHlpNetworkParams
TNetworkParams = record
  HostName: string ;
  DomainName: string ;
  CurrentDnsServer: string ;
  DnsServerTot: integer ;
  DnsServerNames: array [0..9] of string ;
  NodeType: UINT;
  ScopeID: string ;
  EnableRouting: UINT;
  EnableProxy: UINT;
  EnableDNS: UINT;
end;

TIfRows = array of TMibIfRow ; // динамічний масив колонок

// для IpHlpAdaptersInfo
TAdaptorInfo = record
  AdapterName: string ;

```



```

function NextToken( var s: string; Separator: char ): string;
var
  Sep_Pos      : byte;
begin
  Result := ' ';
  if length( s ) > 0 then begin
    Sep_Pos := pos( Separator, s );
    if Sep_Pos > 0 then begin
      Result := copy( s, 1, Pred( Sep_Pos ) );
      Delete( s, 1, Sep_Pos );
    end
    else begin
      Result := s;
      s := ' ';
    end;
  end;
end;

//-----
{ перетворення числового MAC-адреса до ww-xx-yy-zz рядка }
function MacAddr2Str( MacAddr: TMacAddress; size: integer ): string;
var
  i      : integer;
begin
  if Size = 0 then
    begin
      Result := ' 00-00-00' ;
      EXIT;
    end
  else Result := ' ';
  //
  for i := 1 to Size do
    Result := Result + IntToHex( MacAddr[i], 2) + ' -' ;
  Delete( Result, Length( Result ), 1 );
end;

//-----
{ перетворення IP-адреси в мережний байт типу DWORD }
function IpAddr2Str( IPAddr: DWORD ): string;
var
  i      : integer;
begin
  Result := ' ';
  for i := 1 to 4 do
    begin
      Result := Result + Format( ' %3d.' , [IPAddr and $FF] );
      IPAddr := IPAddr shr 8;
    end;
  Delete( Result, Length( Result ), 1 );
end;

//-----
{ перетворення крапкової десяткової IP-адреси в мережний байт типу DWORD}
function Str2IpAddr( IPStr: string ): DWORD;
var
  i      : integer;
  Num    : DWORD;
begin
  Result := 0;
  for i := 1 to 4 do
    try
      Num := ( StrToInt( NextToken( IPStr, ' .' ) ) ) shl 24;
      Result := ( Result shr 8 ) or Num;
    except
      Result := 0;
    end;
  end;
end;

```

```

//-----
{ перетворення номеру порту в мережний байт типу DWORD }
function Port2Wrd( nwoPort: DWORD ): DWORD;
begin
  Result := Swap( WORD( nwoPort ) );
end;

//-----
{ перетворення номеру порту в мережний байт типу string }
function Port2Str( nwoPort: DWORD ): string;
begin
  Result := IntToStr( Port2Wrd( nwoPort ) );
end;

//-----
{ перетворення номеру порту в сервіс ID }
function Port2Svc( Port: DWORD ): string;
var
  i          : integer;
begin
  Result := Format( '%4d', [Port] ); // у випадку, якщо порт не знайдено
  for i := Low( WellKnownPorts ) to High( WellKnownPorts ) do
    if Port = WellKnownPorts[i].Prt then
      begin
        Result := WellKnownPorts[i].Srv;
        BREAK;
      end;
  end;
end;

//-----
{ голова частина, фіксація мережних параметрів }

procedure Get_NetworkParams( List: TStrings );
var
  NetworkParams: TNetworkParams ;
  I, ErrorCode: integer ;
begin
  if not Assigned( List ) then EXIT;
  List.Clear;
  ErrorCode := IpHlpNetworkParams (NetworkParams) ;
  if ErrorCode <> 0 then
    begin
      List.Add (SysErrorMessage (ErrorCode));
      exit;
    end ;
  with NetworkParams do
    begin
      List.Add( ' Ім'я хосту          : ' + HostName );
      List.Add( ' Домен              : ' + DomainName );
      List.Add( ' NETBIOS тип : ' + NETBIOSTypes[NodeType] );
      List.Add( ' DHCP область       : ' + ScopeID );
      List.Add( ' ROUTING визначено  : ' + IntToStr( EnableRouting ) );
      List.Add( ' PROXY визначено   : ' + IntToStr( EnableProxy ) );
      List.Add( ' DNS визначено     : ' + IntToStr( EnabledDNS ) );
      if DnsServerTot <> 0 then
        begin
          for I := 0 to Pred (DnsServerTot) do
            List.Add( ' DNS адреса серверу : ' + DnsServerNames [I] );
          end ;
        end ;
    end ;
end ;

//-----//
function IpHlpNetworkParams (var NetworkParams: TNetworkParams): integer ;
var
  FixedInfo      : PTFixedInfo;          // данні
  InfoSize       : Longint;
  PDnsServer     : PTIP_ADDR_STRING ;    // данні
begin

```

```

InfoSize := 0 ; // данні
result := ERROR_NOT_SUPPORTED ;
if NOT LoadIpHlp then exit ;
result := GetNetworkParams( Nil, @InfoSize ) ; // данні
if result <> ERROR_BUFFER_OVERFLOW then exit ; // данні
GetMem (FixedInfo, InfoSize) ; // данні
try
result := GetNetworkParams( FixedInfo, @InfoSize ) ; // данні
if result <> ERROR_SUCCESS then exit ;
NetworkParams.DnsServerTot := 0 ;
with FixedInfo^ do
begin
NetworkParams.HostName := trim (HostName) ;
NetworkParams.DomainName := trim (DomainName) ;
NetworkParams.ScopeId := trim (ScopeID) ;
NetworkParams.NodeType := NodeType ;
NetworkParams.EnableRouting := EnableRouting ;
NetworkParams.EnableProxy := EnableProxy ;
NetworkParams.EnableDNS := EnabledDNS ;
NetworkParams.DnsServerNames [0] := DNSServerList.IPAddress ; // данні
if NetworkParams.DnsServerNames [0] <> ` ` then
NetworkParams.DnsServerTot := 1 ;
PDnsServer := DnsServerList.Next;
while PDnsServer <> Nil do
begin
NetworkParams.DnsServerNames [NetworkParams.DnsServerTot] :=
PDnsServer^.IPAddress ; // данні
inc (NetworkParams.DnsServerTot) ;
if NetworkParams.DnsServerTot >=
Length (NetworkParams.DnsServerNames) then exit ;
PDnsServer := PDnsServer.Next ;
end;
end ;
finally
FreeMem (FixedInfo) ; // данні
end ;
end;

//-----

function ICMPErr2Str( ICMPErrCode: DWORD) : string;
begin
Result := ` UnknownError : ` + IntToStr( ICMPErrCode ) ;
dec( ICMPErrCode, ICMP_ERROR_BASE );
if ICMPErrCode in [Low(ICMPerr)..High(ICMPerr)] then
Result := ICMPerr[ ICMPErrCode];
end;

//-----

// включення байтів у/з для кожного адаптера

function IpHlpIfTable(var IfTot: integer; var IfRows: TIfRows): integer ;
var
I,
TableSize : integer;
pBuf, pNext : PChar;
begin
result := ERROR_NOT_SUPPORTED ;
if NOT LoadIpHlp then exit ;
SetLength (IfRows, 0) ;
IfTot := 0 ; // данні
TableSize := 0;
// перший виклик: необхідно отримати розмір пам' яті
result := GetIfTable (Nil, @TableSize, false) ; // данні
if result <> ERROR_INSUFFICIENT_BUFFER then exit ;
GetMem( pBuf, TableSize );
try

```

```

FillChar (pBuf^, TableSize, #0); // очищаємо буфер з W98 не беремо
кранку таблиці
result := GetIfTable (PTMibIfTable (pBuf), @TableSize, false) ;
if result <> NO_ERROR then exit ;
IfTot := PTMibIfTable (pBuf)^.dwNumEntries ;
if IfTot = 0 then exit ;
SetLength (IfRows, IfTot) ;
pNext := pBuf + SizeOf(IfTot) ;
for i := 0 to Pred (IfTot) do
begin
    IfRows [i] := PTMibIfRow (pNext )^ ;
    inc (pNext, SizeOf (TMibIfRow)) ;
end;
finally
    FreeMem (pBuf) ;
end ;
end;

procedure Get_IfTable( List: TStrings );
var
    IfRows      : TIfRows ;
    Error, I     : integer;
    NumEntries  : integer;
    sDescr, sIfName: string ;
begin
    if not Assigned( List ) then EXIT;
    List.Clear;
    SetLength (IfRows, 0) ;
    Error := IpHlpIfTable (NumEntries, IfRows) ;
    if (Error <> 0) then
        List.Add( SysErrorMessage( GetLastError ) )
    else if NumEntries = 0 then
        List.Add( ' даних немає ' )
    else
        begin
            for I := 0 to Pred (NumEntries) do
                begin
                    with IfRows [I] do
                        begin
                            if wszName [1] = #0 then
                                sIfName := ' '
                            else
                                sIfName := WideCharToString (@wszName) ; // конвертуємо Юнікод
до рядка
                                sIfName := trim (sIfName) ;
                                sDescr := bDescr ;
                                sDescr := trim (sDescr);
                                List.Add (Format (
                                    ' %0.8x |%3d | %16s |%8d |%12d |%2d |%2d |%10d |%10d | %-s| %-s'
                                ,
                                    [dwIndex, dwType, MacAddr2Str( TMacAddress( bPhysAddr ) ,
                                        dwPhysAddrLen ), dwMTU, dwSpeed, dwAdminStatus,
                                        dwOperStatus, Int64 (dwInOctets), Int64 (dwOutOctets), //
конвертуємо до 32-біт
                                        sIfName, sDescr] ) // данні, додані в/з
                                    );
                                end;
                            end ;
                        end ;
                    SetLength (IfRows, 0) ; // вільна пам' ять
                end ;
            end ;

function IpHlpIfEntry(Index: integer; var IfRow: TMibIfRow): integer ;
begin
    result := ERROR_NOT_SUPPORTED ;
    if NOT LoadIpHlp then exit ;
    FillChar (IfRow, SizeOf (TMibIfRow), #0); // очищаємо буфер з W98 не беремо
    IfRow.dwIndex := Index ;
    result := GetIfEntry (@IfRow) ;
end ;

```

```

end ;

//-----
{ інформація про інсталювані адаптери }

function IpHlpAdaptersInfo(var AdpTot: integer; var AdpRows: TAdaptorRows):
integer ;
var
  BufLen      : DWORD;
  AdapterInfo  : PTIP_ADAPTER_INFO;
  PIPAddr     : PTIP_ADDR_STRING;
  PBuf        : PCHAR ;
  I           : integer ;
begin
  SetLength (AdpRows, 4) ;
  AdpTot := 0 ;
  BufLen := 0 ;
  result := GetAdaptersInfo( Nil, @BufLen );
  if (result <> ERROR_INSUFFICIENT_BUFFER) and (result = NO_ERROR) then exit ;
  GetMem( pBuf, BufLen );
  try
    FillChar (pBuf^, BufLen, #0); // очищуємо буфер
    result := GetAdaptersInfo( PTIP_ADAPTER_INFO (PBuf), @BufLen );
    if result = NO_ERROR then
      begin
        AdapterInfo := PTIP_ADAPTER_INFO (PBuf) ;
        while ( AdapterInfo <> nil ) do
          begin
            AdpRows [AdpTot].IPAddressTot := 0 ;
            SetLength (AdpRows [AdpTot].IPAddressList, 2) ;
            SetLength (AdpRows [AdpTot].IPMaskList, 2) ;
            AdpRows [AdpTot].GatewayTot := 0 ;
            SetLength (AdpRows [AdpTot].GatewayList, 2) ;
            AdpRows [AdpTot].DHCPTot := 0 ;
            SetLength (AdpRows [AdpTot].DHCPSTotal, 2) ;
            AdpRows [AdpTot].PrimWINSTot := 0 ;
            SetLength (AdpRows [AdpTot].PrimWINSServer, 2) ;
            AdpRows [AdpTot].SecWINSTot := 0 ;
            SetLength (AdpRows [AdpTot].SecWINSServer, 2) ;
            AdpRows [AdpTot].CurrIPAddress := NULL_IP;
            AdpRows [AdpTot].CurrIPMask := NULL_IP;
            AdpRows [AdpTot].AdapterName := Trim( string(
AdapterInfo^.AdapterName ) );
            AdpRows [AdpTot].Description := Trim( string(
AdapterInfo^.Description ) );
            AdpRows [AdpTot].MacAddress := MacAddr2Str( TMacAddress(
AdapterInfo^.Address ) ,
AdapterInfo^.AddressLength ) ;
            AdpRows [AdpTot].Index := AdapterInfo^.Index ;
            AdpRows [AdpTot].aType := AdapterInfo^.aType ;
            AdpRows [AdpTot].DHCPEnabled := AdapterInfo^.DHCPEnabled ;
            if AdapterInfo^.CurrentIPAddress <> Nil then
              begin
                AdpRows [AdpTot].CurrIPAddress :=
AdapterInfo^.CurrentIPAddress.IpAddress ;
                AdpRows [AdpTot].CurrIPMask :=
AdapterInfo^.CurrentIPAddress.IpMask ;
              end ;

            // беремо список IP адрес та конвертуємо в IPAddressList
            I := 0 ;
            PIPAddr := @AdapterInfo^.IPAddressList ;
            while (PIPAddr <> Nil) do
              begin
                AdpRows [AdpTot].IPAddressList [I] := PIPAddr.IpAddress ;
                AdpRows [AdpTot].IPMaskList [I] := PIPAddr.IpMask ;
                PIPAddr := PIPAddr.Next ;
                inc (I) ;
                if Length (AdpRows [AdpTot].IPAddressList) <= I then

```

```

begin
    SetLength (AdpRows [AdpTot].IPAddressList, I -2) ;
    SetLength (AdpRows [AdpTot].IPMaskList, I -2) ;
end ;
end ;
AdpRows [AdpTot].IPAdressTot := I ;

// беремо список IP адрес для GatewayList
I := 0 ;
PIpAddr := @AdapterInfo^.GatewayList ;
while (PIpAddr <> Nil) do
begin
    AdpRows [AdpTot].GatewayList [I] := PIpAddr.IpAddress ;
    PIpAddr := PIpAddr.Next ;
    inc (I) ;
    if Length (AdpRows [AdpTot].GatewayList) <= I then
        SetLength (AdpRows [AdpTot].GatewayList, I -2) ;
    end ;
    AdpRows [AdpTot].GatewayTot := I ;

// беремо список IP адрес для GatewayList
I := 0 ;
PIpAddr := @AdapterInfo^.DHCPSTotal ;
while (PIpAddr <> Nil) do
begin
    AdpRows [AdpTot].DHCPSTotal [I] := PIpAddr.IpAddress ;
    PIpAddr := PIpAddr.Next ;
    inc (I) ;
    if Length (AdpRows [AdpTot].DHCPSTotal) <= I then
        SetLength (AdpRows [AdpTot].DHCPSTotal, I -2) ;
    end ;
    AdpRows [AdpTot].DHCPTot := I ;

// беремо список IP адрес для PrimaryWINSServer
I := 0 ;
PIpAddr := @AdapterInfo^.PrimaryWINSServer ;
while (PIpAddr <> Nil) do
begin
    AdpRows [AdpTot].PrimWINSServer [I] := PIpAddr.IpAddress ;
    PIpAddr := PIpAddr.Next ;
    inc (I) ;
    if Length (AdpRows [AdpTot].PrimWINSServer) <= I then
        SetLength (AdpRows [AdpTot].PrimWINSServer, I -2) ;
    end ;
    AdpRows [AdpTot].PrimWINSTot := I ;

// беремо список IP адрес для SecondaryWINSServer
I := 0 ;
PIpAddr := @AdapterInfo^.SecondaryWINSServer ;
while (PIpAddr <> Nil) do
begin
    AdpRows [AdpTot].SecWINSServer [I] := PIpAddr.IpAddress ;
    PIpAddr := PIpAddr.Next ;
    inc (I) ;
    if Length (AdpRows [AdpTot].SecWINSServer) <= I then
        SetLength (AdpRows [AdpTot].SecWINSServer, I -2) ;
    end ;
    AdpRows [AdpTot].SecWINSTot := I ;

    AdpRows [AdpTot].LeaseObtained := AdapterInfo^.LeaseObtained ;
    AdpRows [AdpTot].LeaseExpires := AdapterInfo^.LeaseExpires ;

    inc (AdpTot) ;
    if Length (AdpRows) <= AdpTot then
        SetLength (AdpRows, AdpTot -2) ; // більше пам' яті
    AdapterInfo := AdapterInfo^.Next ;
end ;
SetLength (AdpRows, AdpTot) ;
end ;

```

```

    finally
        FreeMem( pBuf );
    end ;
end ;

procedure Get_AdaptersInfo( List: TStrings );
var
    AdpTot: integer;
    AdpRows: TAdaptorRows ;
    Error: DWORD ;
    I: integer ;
    //J: integer ;
    //S: string ;          id.
begin
    if not Assigned( List ) then EXIT;
    List.Clear;
    SetLength (AdpRows, 0) ;
    AdpTot := 0 ;
    Error := IpHlpAdaptersInfo(AdpTot, AdpRows) ;
    if (Error <> 0) then
        List.Add( SysErrorMessage( GetLastError ) )
    else if AdpTot = 0 then
        List.Add( ' дaнних немає ' )
    else
        begin
            for I := 0 to Pred (AdpTot) do
                begin
                    with AdpRows [I] do
                        begin
                            //List.Add(AdapterName + ' |' + Description ); // jpt : не
                            //використовується
                            List.Add( Format( ' %8.8x | %6s | %16s | %2d | %16s | %16s | %16s' ,
                                [Index, AdaptTypes[aType], MacAddress, DHCPEnabled,
                                GatewayList [0], DHCPServer [0], PrimWINSServer [0]] ) );
                            {if IPAddressTot <> 0 then // jpt : не використовується
                                begin
                                    S := ' ' ;
                                    for J := 0 to Pred (IPAddressTot) do
                                        S := S + IPAddressList [J] + ' /' + IPMaskList [J] + '
                                | ' ;
                                    List.Add(IntToStr (IPAddressTot) + ' IP Adresse(s): ' + S);
                                end ;
                                List.Add( ' ' ); }
                            end ;
                        end ;
                    end ;
                end ;
            SetLength (AdpRows, 0) ;
        end ;

//-----
{ моні торимо час доступу до IP }
function Get_RTTAndHopCount( IPAddr: DWORD; MaxHops: Longint; var RTT: Longint;
    var HopCount: Longint ): integer;
begin
    if not GetRTTAndHopCount( IPAddr, @HopCount, MaxHops, @RTT ) then
        begin
            Result := GetLastError;
            RTT :=-1; // Расположення BAD_HOST_NAME,etc...
            HopCount :=-1;
        end
    else
        Result := NO_ERROR;
    end;

//-----
{ ARP-таблиця включає відношення між віддаленим IP та віддаленим MAC-адресом.
}
procedure Get_ARPTable( List: TStrings );
var

```

```

IPNetRow      : TMibIPNetRow;
TableSize     : DWORD;
NumEntries    : DWORD;
ErrorCode     : DWORD;
i             : integer;
pBuf         : PChar;
begin
  if not Assigned( List ) then EXIT;
  List.Clear;
  // перший виклик: беремо довжину таблиці
  TableSize := 0;
  ErrorCode := GetIPNetTable( Nil, @TableSize, false ); // данні
  //
  if ErrorCode = ERROR_NO_DATA then
  begin
    List.Add( ' ARP-кеш пустий.' );
    EXIT;
  end;
  // беремо таблицю
  GetMem( pBuf, TableSize );
  NumEntries := 0 ;
  try
    ErrorCode := GetIpNetTable( PTMIBIPNetTable( pBuf ), @TableSize, false );
    if ErrorCode = NO_ERROR then
    begin
      NumEntries := PTMIBIPNetTable( pBuf )^.dwNumEntries;
      if NumEntries > 0 then
      begin
        inc( pBuf, SizeOf( DWORD ) ); // беремо розмір останньої таблиці
        for i := 1 to NumEntries do
        begin
          IPNetRow := PTMIBIPNetRow( PBuf )^;
          with IPNetRow do
            List.Add( Format( ' %8x | %12s | %16s | %10s' ,
              [dwIndex, MacAddr2Str( bPhysAddr, dwPhysAddrLen ),
                IPAddr2Str( dwAddr ), ARPEntryType[dwType]
              ]));
            inc( pBuf, SizeOf( IPNetRow ) );
          end;
        end
      end
    else
      List.Add( ' ARP-кеш пустий.' );
    end
  else
    List.Add( SysErrorMessage( ErrorCode ) );

    // необхідно відновити показник!
  finally
    dec( pBuf, SizeOf( DWORD ) + NumEntries -SizeOf( IPNetRow ) );
    FreeMem( pBuf );
  end ;
end;

//-----
procedure Get_TCPTable( List: TStrings );
var
  TCPRow      : TMIBTCPRow;
  i,
  NumEntries  : integer;
  TableSize   : DWORD;
  ErrorCode   : DWORD;
  DestIP      : string;
  pBuf        : PChar;
begin
  if not Assigned( List ) then EXIT;
  List.Clear;
  RecentIPs.Clear;
  // перший виклик: беремо довжину таблиці

```

```

TableSize := 0;
NumEntries := 0 ;
ErrorCode := GetTCPTable( Nil, @TableSize, false ); // данні
if ErrorCode <> ERROR_INSUFFICIENT_BUFFER then
    EXIT;

// беремо розмір пам'яті, викликаємо знову
GetMem( pBuf, TableSize );
// беремо таблицю
ErrorCode := GetTCPTable( PTMIBTCPTable( pBuf ), @TableSize, false );
if ErrorCode = NO_ERROR then
begin

    NumEntries := PTMIBTCPTable( pBuf )^.dwNumEntries;
    if NumEntries > 0 then
    begin
        inc( pBuf, SizeOf( DWORD ) ); // беремо розмір останньої таблиці
        for i := 1 to NumEntries do
        begin
            TCPRow := PTMIBTCPRow( pBuf )^; // беремо останній запис
            with TCPRow do
            begin
                if dwRemoteAddr = 0 then
                    dwRemotePort := 0;
                DestIP := IPAddr2Str( dwRemoteAddr );
                List.Add(
                    Format( ' %15s : %-7s | %15s : %-7s | %-16s' ,
                        [IpAddr2Str( dwLocalAddr ),
                          Port2Svc( Port2Wrd( dwLocalPort ) ),
                          DestIP,
                          Port2Svc( Port2Wrd( dwRemotePort ) ),
                          TCPConnState[dwState]
                        ] ) );
                //
                if ( not ( dwRemoteAddr = 0 ) )
                    and ( RecentIps.IndexOf( DestIP ) = -1 ) then
                    RecentIps.Add( DestIP );
            end;
            inc( pBuf, SizeOf( TMIBTCPRow ) );
        end;
    end;
end
else
    List.Add( SysErrorMessage( ErrorCode ) );
dec( pBuf, SizeOf( DWORD ) + NumEntries -SizeOf( TMibTCPRow ) );
FreeMem( pBuf );
end;

//-----
procedure Get_TCPStatistics( List: TStrings );
var
    TCPStats      : TMibTCPStats;
    ErrorCode      : DWORD;
begin
    if not Assigned( List ) then EXIT;
    List.Clear;
    if NOT LoadIpHlp then exit ;
    ErrorCode := GetTCPStatistics( @TCPStats );
    if ErrorCode = NO_ERROR then
        with TCPStats do
        begin
            List.Add( ' Алгоритм повторної передачі : ' + TCPToAlgo[dwRTOAlgorithm]
                );
            List.Add( ' Мінімальний час виходу          : ' + IntToStr( dwRTOMin ) + '
                ms' );
            List.Add( ' Максимальний час виходу          : ' + IntToStr( dwRTOMax ) + '
                ms' );
            List.Add( ' Максимальне число підключень : ' + IntToStr( dwRTOAlgorithm )
                );
        end;
    end;
end;

```

```

        List.Add( ' Активні підключення          : ' + IntToStr( dwActiveOpens
    ) );
    List.Add( ' пасивні підключення          : ' + IntToStr( dwPassiveOpens
    ) );
    List.Add( ' Невдала спроба відкриття      : ' + IntToStr( dwAttemptFails )
    );
    List.Add( ' Скидання встановленого підключення : ' + IntToStr(
dwEstabResets ) );
    List.Add( ' Поточне встановлене підключення.: ' + IntToStr( dwCurrEstab )
    );
    List.Add( ' Отримані сегменти              : ' + IntToStr( dwInSegs ) );
    List.Add( ' Передані сегменти              : ' + IntToStr( dwOutSegs ) );
    List.Add( ' Перепідключені сегменти       : ' + IntToStr( dwReTransSegs ) );
    List.Add( ' помилка входу                  : ' + IntToStr( dwInErrs ) );
    List.Add( ' Перезавантаження вихідних     : ' + IntToStr( dwOutRsts
    ) );
    List.Add( ' Сумарні зв'язки                : ' + IntToStr( dwNumConns ) );
end
else
    List.Add( SysErrorMessage( ErrorCode ) );
end;

function IpHlpTCPStatistics (var TCPStats: TMibTCPStats): integer ;
begin
    result := ERROR_NOT_SUPPORTED ;
    if NOT LoadIpHlp then exit ;
    result := GetTCPStatistics( @TCPStats );
end;

//-----
procedure Get_UDPTable( List: TStrings );
var
    UDPRow      : TMIBUDPRow;
    i,
    NumEntries  : integer;
    TableSize   : DWORD;
    ErrorCode   : DWORD;
    pBuf        : PChar;
begin
    if not Assigned( List ) then EXIT;
    List.Clear;

    // перший виклик: беремо довжину таблиці
    TableSize := 0;
    NumEntries := 0 ;
    ErrorCode := GetUDPTable( Nil, @TableSize, false );
    if ErrorCode <> ERROR_INSUFFICIENT_BUFFER then
        EXIT;

    // виділяємо пам'ять, викликаємо знову
    GetMem( pBuf, TableSize );

    // беремо таблицю
    ErrorCode := GetUDPTable( PTMIBUDPTable( pBuf ), @TableSize, false );
    if ErrorCode = NO_ERROR then
        begin
            NumEntries := PTMIBUDPTable( pBuf )^.dwNumEntries;
            if NumEntries > 0 then
                begin
                    inc( pBuf, SizeOf( DWORD ) ); // беремо розмір останньої таблиці
                    for i := 1 to NumEntries do
                        begin
                            UDPRow := PTMIBUDPRow( pBuf )^; // беремо останній запис
                            with UDPRow do
                                List.Add( Format( ' %15s : %-6s' ,
                                    [IpAddr2Str( dwLocalAddr ),
                                    Port2Svc( Port2Wrd( dwLocalPort ) )
                                    ] ) );
                            inc( pBuf, SizeOf( TMIBUDPRow ) );
                        end;
                    end;
                end;
        end;
    end;
end;

```

```

        end;
    end
    else
        List.Add( ' немає даних.' );
    end
    else
        List.Add( SysErrorMessage( ErrorCode ) );
    dec( pBuf, SizeOf( DWORD ) + NumEntries -SizeOf( TMibUDPRow ) );
    FreeMem( pBuf );
end;

//-----
procedure Get_IPAddrTable( List: TStrings );
var
    IPAddrRow      : TMibIPAddrRow;
    TableSize      : DWORD;
    ErrorCode       : DWORD;
    i               : integer;
    pBuf           : PChar;
    NumEntries     : DWORD;
begin
    if not Assigned( List ) then EXIT;
    List.Clear;
    TableSize := 0; ;
    NumEntries := 0 ;
    // перший виклик: беремо довжину таблиці
    ErrorCode := GetIpAddrTable( Nil, @TableSize, true ); // данні
    if ErrorCode <> ERROR_INSUFFICIENT_BUFFER then
        EXIT;

    GetMem( pBuf, TableSize );
    // беремо таблицю
    ErrorCode := GetIpAddrTable( PTMibIPAddrTable( pBuf ), @TableSize, true );
    if ErrorCode = NO_ERROR then
        begin
            NumEntries := PTMibIPAddrTable( pBuf )^.dwNumEntries;
            if NumEntries > 0 then
                begin
                    inc( pBuf, SizeOf( DWORD ) );
                    for i := 1 to NumEntries do
                        begin
                            IPAddrRow := PTMIBIPAddrRow( pBuf )^;
                            with IPAddrRow do
                                List.Add( Format( ' %8.8x | %15s | %15s | %15s | %8.8d' ,
                                    [dwIndex,
                                    IPAddr2Str( dwAddr ),
                                    IPAddr2Str( dwMask ),
                                    IPAddr2Str( dwBCastAddr ),
                                    dwReasmSize
                                    ] ) );
                                inc( pBuf, SizeOf( TMIBIPAddrRow ) );
                            end;
                        end
                    else
                        List.Add( ' немає даних.' );
                    end
                end
            else
                List.Add( SysErrorMessage( ErrorCode ) );
            end;

            // відновлюємо показчик!
            dec( pBuf, SizeOf( DWORD ) + NumEntries -SizeOf( IPAddrRow ) );
            FreeMem( pBuf );
        end;

//-----
{ отримуємо дані з таблиці маршрутизації; }
procedure Get_IPForwardTable( List: TStrings );
var
    IPForwRow      : TMibIPForwardRow;

```

```

TableSize      : DWORD;
ErrorCode      : DWORD;
i              : integer;
pBuf           : PChar;
NumEntries     : DWORD;
begin

    if not Assigned( List ) then EXIT;
    List.Clear;
    TableSize := 0;

    // перший виклик: беремо довжину таблиці
    NumEntries := 0 ;
    ErrorCode := GetIpForwardTable( Nil, @TableSize, true);
    if ErrorCode <> ERROR_INSUFFICIENT_BUFFER then
        EXIT;

    // беремо таблицю
    GetMem( pBuf, TableSize );
    ErrorCode := GetIpForwardTable( PTMibIPForwardTable( pBuf ), @TableSize,
true);
    if ErrorCode = NO_ERROR then
        begin
            NumEntries := PTMibIPForwardTable( pBuf )^.dwNumEntries;
            if NumEntries > 0 then
                begin
                    inc( pBuf, SizeOf( DWORD ) );
                    for i := 1 to NumEntries do
                        begin
                            IPForwRow := PTMibIPForwardRow( pBuf )^;
                            with IPForwRow do
                                begin
                                    if (dwForwardType < 1)
                                        or (dwForwardType > 4) then
                                        dwForwardType := 1 ; // данні
                                    List.Add( Format(
                                        ' %15s | %15s | %15s | %8.8x | %7s | %5.5d | %7s | %2.2d' ,
                                        [IPAddr2Str( dwForwardDest ),
                                        IPAddr2Str( dwForwardMask ),
                                        IPAddr2Str( dwForwardNextHop ),
                                        dwForwardIFIndex,
                                        IPForwTypes[dwForwardType],
                                        dwForwardNextHopAS,
                                        IPForwProtos[dwForwardProto],
                                        dwForwardMetric1
                                        ] ) );
                                    end ;
                                    inc( pBuf, SizeOf( TMibIPForwardRow ) );
                                end;
                            end
                        else
                            List.Add( ' немає даних.' );
                        end
                    else
                        List.Add( SysErrorMessage( ErrorCode ) );
                    dec( pBuf, SizeOf( DWORD ) + NumEntries -SizeOf( TMibIPForwardRow ) );
                    FreeMem( pBuf );
                end;
            end;

            //-----
            procedure Get_IPStatistics( List: TStrings );
            var
                IPStats      : TMibIPStats;
                ErrorCode     : integer;
            begin
                if not Assigned( List ) then EXIT;
                if NOT LoadIpHlp then exit ;
                ErrorCode := GetIPStatistics( @IPStats );
                if ErrorCode = NO_ERROR then

```

```

begin
  List.Clear;
  with IPStats do
  begin
    if dwForwarding = 1 then
      List.add( ' Розблокована пересилка      : ' + ' так' )
    else
      List.add( ' Розблокована пересилка      : ' + ' ні' );
      List.add( ' Любий TTL                    : ' + inttostr( dwDefaultTTL ) );
      List.add( ' Датаграма прийнята          : ' + inttostr( dwInReceives ) );
      List.add( ' Помилка заголовку           (In) : ' + inttostr( dwInHdrErrors )
    );
      List.add( ' Помилка адреси              (In) : ' + inttostr( dwInAddrErrors ) );
      List.add( ' Датаграма переслана         : ' + inttostr( dwForwDatagrams ) );
    // данні
      List.add( ' Невизначений протокол (In) : ' + inttostr( dwInUnknownProtos
    ) );
      List.add( ' Датаграма відмовлена        : ' + inttostr( dwInDiscards ) );
      List.add( ' Датаграма встановлена       : ' + inttostr( dwInDelivers ) );
      List.add( ' Зовнішній запит            : ' + inttostr( dwOutRequests )
    );
      List.add( ' Маршрутизація не виконана    : ' + inttostr(
dwRoutingDiscards ) );
      List.add( ' Немає маршрутів             (Out) : ' + inttostr( dwOutNoRoutes )
    );
      List.add( ' Перебраний час              : ' + inttostr( dwReasmTimeOut ) );
      List.add( ' Запит перебору              : ' + inttostr( dwReasmReqds ) );
      List.add( ' Повний перебор              : ' + inttostr( dwReasmOKs ) );
      List.add( ' Помилка перебору           : ' + inttostr( dwReasmFails ) );
      List.add( ' Повна фрагментація         : ' + inttostr( dwFragOKs ) );
      List.add( ' Помилка фрагментації       : ' + inttostr( dwFragFails ) );
      List.add( ' Датаграма фрагментована    : ' + inttostr( dwFRagCreates )
    );
      List.add( ' Кількість інтерфейсів       : ' + inttostr( dwNumIf ) );
      List.add( ' Кількість IP-адрес        : ' + inttostr( dwNumAddr ) );
      List.add( ' Маршрут в таблиці маршрутизатора : ' + inttostr( dwNumRoutes
    ) );
    end;
  end
  else
    List.Add( SysErrorMessage( ErrorCode ) );
  end;

function IpHlpIPStatistics (var IPStats: TMibIPStats): integer ;      // данні
begin
  result := ERROR_NOT_SUPPORTED ;
  if NOT LoadIpHlp then exit ;
  result := GetIPStatistics( @IPStats );
end ;

//-----
procedure Get_UdpStatistics( List: TStrings );
var
  UdpStats      : TMibUDPStats;
  ErrorCode     : integer;
begin
  if not Assigned( List ) then EXIT;
  ErrorCode := GetUDPStatistics( @UdpStats );
  if ErrorCode = NO_ERROR then
  begin
    List.Clear;
    with UDPStats do
    begin
      List.add( ' Датаграми (In)           : ' + inttostr( dwInDatagrams ) );
      List.add( ' Датаграми (Out)          : ' + inttostr( dwOutDatagrams ) );
      List.add( ' Немає портів              : ' + inttostr( dwNoPorts ) );
      List.add( ' Помилка                  (In) : ' + inttostr( dwInErrors ) );
      List.add( ' UDP список портів       : ' + inttostr( dwNumAddrs ) );
    end;
  end;
end;

```

```

end
else
    List.Add( SysErrorMessage( ErrorCode ) );
end;

//-----*//
function IpHlpUdpStatistics (UdpStats: TMibUDPStats): integer ;    // данні
begin
    result := ERROR_NOT_SUPPORTED ;
    if NOT LoadIpHlp then exit ;
    result := GetUDPStatistics (@UdpStats) ;
end ;

//-----
procedure Get_ICMPStats( ICMPIn, ICMPOut: TStrings ) ;
var
    ErrorCode      : DWORD;
    ICMPStats      : PTMibICMPInfo;
begin
    if ( ICMPIn = nil ) or ( ICMPOut = nil ) then EXIT;
    ICMPIn.Clear;
    ICMPOut.Clear;
    New( ICMPStats );
    ErrorCode := GetICMPStatistics( ICMPStats );
    if ErrorCode = NO_ERROR then
        begin
            with ICMPStats.InStats do
                begin
                    ICMPIn.Add( ' Прийнято повідомлень      : ' + IntToStr( dwMsgs ) );
                    ICMPIn.Add( ' Помилка                  : ' + IntToStr( dwErrors ) );
                    ICMPIn.Add( ' Розташування недосягнено   : ' + IntToStr( dwDestUnreachs
                ) );
                    ICMPIn.Add( ' Час перевищений      : ' + IntToStr( dwTimeEcxcds ) );
                    ICMPIn.Add( ' Проблеми з параметрами   : ' + IntToStr( dwParmProbs
                ) );
                    ICMPIn.Add( ' Джерело відключено      : ' + IntToStr( dwSrcQuenchs ) );
                    ICMPIn.Add( ' Переназначено        : ' + IntToStr( dwRedirects ) );
                    ICMPIn.Add( ' Ехо запит           : ' + IntToStr( dwEchos ) );
                    ICMPIn.Add( ' Ехо відповідь        : ' + IntToStr( dwEchoReps ) );
                    ICMPIn.Add( ' Запит мітки часу      : ' + IntToStr( dwTimeStamps ) );
                    ICMPIn.Add( ' Відповідь мітки часу   : ' + IntToStr( dwTimeStampReps
                ) );
                    ICMPIn.Add( ' Запит маски адрес : ' + IntToStr( dwAddrMasks ) );
                    ICMPIn.Add( ' Відповідь маски адрес  : ' + IntToStr( dwAddrReps ) );
                end;
            end;
            //
            with ICMPStats.OutStats do
                begin
                    ICMPOut.Add( ' Повідомлення вдправлено      : ' + IntToStr( dwMsgs ) );
                    ICMPOut.Add( ' Помилка                  : ' + IntToStr( dwErrors ) );
                    ICMPOut.Add( ' Розташування недосягнено   : ' + IntToStr( dwDestUnreachs
                ) );
                    ICMPOut.Add( ' Час перевищений      : ' + IntToStr( dwTimeEcxcds ) );
                    ICMPOut.Add( ' Проблеми з параметрами   : ' + IntToStr( dwParmProbs
                ) );
                    ICMPOut.Add( ' Джерело відключено      : ' + IntToStr( dwSrcQuenchs ) );
                    ICMPOut.Add( ' Переназначено        : ' + IntToStr( dwRedirects ) );
                    ICMPOut.Add( ' Ехо запит           : ' + IntToStr( dwEchos ) );
                    ICMPOut.Add( ' Ехо відповідь        : ' + IntToStr( dwEchoReps ) );
                    ICMPOut.Add( ' Запит мітки часу      : ' + IntToStr( dwTimeStamps ) );
                    ICMPOut.Add( ' Відповідь мітки часу   : ' + IntToStr( dwTimeStampReps
                ) );
                    ICMPOut.Add( ' Запит маски адрес : ' + IntToStr( dwAddrMasks ) );
                    ICMPOut.Add( ' Відповідь маски адрес  : ' + IntToStr( dwAddrReps ) );
                end;
            end;
        end;
    else
        IcmpIn.Add( SysErrorMessage( ErrorCode ) );
        Dispose( ICMPStats );
    end;
end;

```

```
end;

//-----
procedure Get_RecentDestIPs( List: TStrings );
begin
  if Assigned( List ) then
    List.Assign( RecentIPs )
  end;

initialization

  RecentIPs := TStringList.Create;

finalization

  RecentIPs.Free;

end.
```

К6П3_2023

Файл TCP_IP.pas- монітор TCP/IP з'єднань

```

unit TCP_IP;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, IPHelper, IpHlpApi, Buttons;

type
  TForm2 = class(TForm)
    StaticText2: TStaticText;
    StaticText3: TStaticText;
    TCPMemo: TMemo;
    UDPMemo: TMemo;
    Timer1: TTimer;
    cbTimer: TCheckBox;
    btRTTI: TSpeedButton;
    SpeedButton1: TSpeedButton;
    edtRTTI: TEdit;
    procedure Timer1Timer(Sender: TObject);
    procedure SpeedButton1Click(Sender: TObject);
    procedure btRTTIClick(Sender: TObject);
    procedure cbRecentIPsClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
    procedure DOIPStuff;
  public
    { Public declarations }
  end;

var
  Form2: TForm2;

implementation

{$R *.dfm}

procedure TForm2.Timer1Timer(Sender: TObject);
begin
  if cbTimer.State = cbCHECKED then
  begin
    Timer1.Enabled := false;
    DoIPStuff;
    Timer1.Enabled := true;
  end;
end;

procedure TForm2.DOIPStuff;
begin
  Get_TCPTable( TCPMemo.Lines );
  Get_UDPTable( UDPMemo.Lines );

end;

procedure TForm2.SpeedButton1Click(Sender: TObject);
begin
  Speedbutton1.Enabled := false;
  DoIPStuff;
  Speedbutton1.Enabled := true;
end;

procedure TForm2.btRTTIClick(Sender: TObject);
var

```

```

IPadr      : dword;
Rtt, HopCount : longint;
Res       : integer;
begin
  btRTTI.Enabled := false;
  Screen.Cursor := crHOURLASS;
  IPadr := Str2IPAddr( edtRTTI.Text );
  Res := Get_RTTAndHopCount( IPadr, 128, RTT, HopCount );
  if Res = NO_ERROR then
    ShowMessage( ' Час запиту '
      + inttostr( rtt ) + ' ms, '
      + inttostr( HopCount )
      + ' hops to : ' + edtRTTI.Text
    )
  else
    ShowMessage( ' Відбулася помилка:' + #13
      + ICMPErr2Str( Res ) );
  btRTTI.Enabled := true;
  Screen.Cursor := crDEFAULT;

end;

procedure TForm2.cbRecentIPsClick(Sender: TObject);
begin
  //edtRTTI.Text := cbRecentIPs.Items[cbRecentIPs.ItemIndex];
end;

procedure TForm2.FormCreate(Sender: TObject);
begin
  if LoadIpHlp then
    begin
      DOIpStuff;
      Timer1.Enabled := true;
    end
  else
    ShowMessage( ' Інтернет помічник DLL не є доступним, або не підтримується'
  ) ;
end;

end.

```

Файл Stat.pas- статистика мережі

```

unit Stat;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, IPHelper, IpHlpApi, Buttons;

type
  TForm3 = class(TForm)
    StaticText7: TStaticText;
    TCPStatMemo: TMemo;
    StaticText5: TStaticText;
    IPStatsMemo: TMemo;
    StaticText12: TStaticText;
    ICMPInMemo: TMemo;
    ICMPOutMemo: TMemo;
    StaticText4: TStaticText;
    UDPStatsMemo: TMemo;
    Timer1: TTimer;
    cbTimer: TCheckBox;
    btRTTI: TSpeedButton;
    edtRTTI: TEdit;
    procedure Timer1Timer(Sender: TObject);
    procedure btRTTIClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
    procedure DOIpStuff;
  public
    { Public declarations }
  end;

var
  Form3: TForm3;

implementation

{$R *.dfm}

procedure TForm3.DOIpStuff;
begin

  Get_TCPStatistics( TCPStatMemo.Lines );
  Get_IPStatistics( IPStatsMemo.Lines );
  Get_UDPStatistics( UDPStatsMemo.Lines );
  Get_ICMPStats( ICMPInMemo.Lines, ICMPOutMemo.Lines );

end;

procedure TForm3.Timer1Timer(Sender: TObject);
begin
  if cbTimer.State = cbCHECKED then
  begin
    Timer1.Enabled := false;
    DoIPStuff;
    Timer1.Enabled := true;
  end;
end;

procedure TForm3.btRTTIClick(Sender: TObject);
var
  IPadr      : dword;
  Rtt, HopCount : longint;
  Res        : integer;

```

```
begin
  btRTTI.Enabled := false;
  Screen.Cursor := crHOURLASS;
  IPadr := Str2IPAddr( edtRTTI.Text );
  Res := Get_RTTAndHopCount( IPadr, 128, RTT, HopCount );
  if Res = NO_ERROR then
    ShowMessage( ' Час запиту '
      + inttostr( rtt ) + ' ms, '
      + inttostr( HopCount )
      + ' hops to : ' + edtRTTI.Text
    )
  else
    ShowMessage( ' Помилка:' + #13
      + ICMPErr2Str( Res ) ) ;
  btRTTI.Enabled := true;
  Screen.Cursor := crDEFAULT;

end;

procedure TForm3.FormCreate(Sender: TObject);
begin
  if LoadIpHlp then
  begin
    DOIpStuff;
    Timer1.Enabled := true;
  end
  else
    ShowMessage( 'Інтернет помічник DLL не є доступним, або не підтримується'
  ) ;
end;

end.
```

Файл About.pas- довідка

```
unit About;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls;

type
  TForm1 = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    Button1: TButton;
    Image2: TImage;
    Image1: TImage;
    Image3: TImage;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.dfm}

procedure TForm1.Button1Click(Sender: TObject);
begin
  Form1.Close;
end;

end.
```