

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 віддаленого
адміністрування комп’ютера у мережі для організації
техпідтримки”

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

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

Кілочицькій Марині Леонідівні

(прізвище, ім'я, по батькові)

1. Тема роботи

Дослідження та програмна реалізація системи віддаленого адміністрування комп’ютера у мережі для організації техпідтримки

2. Керівник роботи

Дресєва Ганна Миколаївна, доктор філософії (PhD)

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

затверджені наказом вищого навчального закладу № 37-13 від 04.08.2023 року

3. Строк подання студентом роботи до захисту

10.12.2023 р.

4. Мета та завдання випускної кваліфікаційної роботи: *Метою розробки є дослідження та програмна реалізація системи віддаленого адміністрування комп’ютера у мережі для організації техпідтримки*

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

1. Призначення та область використання.

6. Наукова новизна.

2. Перегляд аналогічних існуючих систем.

7. Економічна ефективність розробленої програми.

3. Опис і обґрунтування проектних рішень.

8. Заходи з охорони праці та техніки безпеки.

4. Етапи програмування системи.

9. Висновки.

5. Впровадження системи в промислову експлуатацію

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

Наукова новизна

1 аркуш

Структурна схема системи

1 аркуш

Функціональна схема системи

1 аркуш

Діаграма процесів

1 аркуш

Блок-схема алгоритму роботи додатку

2 аркуша

Показники економічної ефективності

1 аркуш

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	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

Кілочичька М.Л. Дослідження та програмна реалізація системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки. 122 Комп'ютерні науки. Центральнотраїнський національний технічний університет. Кропивницький. 2023.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки.

Метою розробки є дослідження та програмна реалізація системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки.

Об'єктом дослідження є процес віддаленого адміністрування комп'ютера у мережі для організації техпідтримки.

Предметом дослідження є методи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки.

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

Результат роботи – програмна реалізація системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки.

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 RAD Studio Delphi.

Ключові слова: комп'ютерні науки, адміністрування комп'ютера

ABSTRACT

Kilochytska M.L. Research and software implementation of a system of remote computer administration in the network for the organization of technical support. 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 system of remote computer administration in the network for the organization of technical support.

The purpose of the development is research and software implementation of a remote computer administration system in the network for the organization of technical support.

The object of the study is the process of remote computer administration in the network for the organization of technical support.

The subject of research is the methods of remote computer administration in the network for the organization of technical support.

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

The result of the work is the software implementation of the remote computer administration system in the network for the organization of technical support.

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 RAD Studio Delphi environment.

Keywords: computer science, computer administration

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	7
1.1 Призначення системи.....	7
1.2 Область застосування.....	8
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	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	22
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	24
3.1 Опис функціонування системи	24
3.2 Розробка структурної схеми.....	27
3.3 Розробка функціональної схеми	42
3.4 Розробка діаграми процесів.....	47
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	49
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	49
4.2 Захист розробленого програмного забезпечення.....	60
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	62
6 НАУКОВА НОВИЗНА	64

					ВКРМ-122.23.0067.00.00.ПЗ			
Вим	Арк	№ докум.	Підп.	Дата	<i>Дослідження та програмна реалізація системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки</i>	Літ.	Аркуш	Аркушів
<i>Розроб.</i>	<i>Кілоцицька М.Л.</i>					М	1	106
<i>Перев.</i>	<i>Дресва Г.М.</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.....	65
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	65
7.2 Розрахунок трудомісткості розробки програмної продукції.....	67
7.3 Визначення чисельності виконавців і планового фонду зарплати.....	69
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника.....	74
7.5 Визначення собівартості розробки та ціни програмної продукції.....	78
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	81
7.7 Визначення експлуатаційних витрат.....	82
7.8 Визначення економічної ефективності програмної продукції.....	83
7.9 Висновок.....	85
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	86
8.1 Вступ.....	86
8.2 Аналіз умов праці	88
8.3 Техніка безпеки та протипожежна профілактика	91
8.4 Розрахункова частина	93
8.5 Список використаних джерел інформації.....	95
9 ОСНОВНІ ВИСНОВКИ.....	98
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	100

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

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

ВСТУП

Актуальність теми. Віддалене керування комп'ютерами для надання віддаленої підтримки сьогодні здобуває все більшу популярність. Це цілком обґрунтовано, тому що має безліч плюсів, наприклад, віддалене керування комп'ютером через Інтернет. Замислюючись про те, чи потрібна вам ця послуга віддалене керування комп'ютером, варто зрозуміти, що вона собою представляє.

У багатьох користувачів комп'ютерів і інших ІТ-систем може виникнути питання доцільності віддаленого керування, про те, наскільки воно вигідно й корисно. При цьому користь від нього очевидна:

1. Виключається необхідність виклику ІТ-персоналу при виникненні неполадок.
2. Виключається можливість раптової зупинки роботи через певні несправності, тому що дана послуга дозволяє здійснювати безперервний онлайн аудит системи й вчасно діагностувати й усувати проблеми по засобах віддаленого робочого стола.
3. Немає необхідності подовгу очікувати фахівця, щоб довідатися як підключитися до комп'ютера друга, тому що при виникненні неполадок фахівці, що здійснюють дистанційний контроль над станом системи, можуть приступитися до їхнього усунення негайно, задіявши remote desktop.
4. Підвищуються загальні результати роботи ІТ-систем в організації.

Це лише основні переваги, який характеризується робочий стіл віддалений доступ. І вже при обліку перерахованих переваг можна зробити вивід, що керування комп'ютером віддалено може стати необхідним при будь-яких умовах.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки.

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

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

- Огляд існуючих систем віддаленого адміністрування комп'ютера у мережі для організації техпідтримки.
- Дослідження системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки.
- Програмна реалізація системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки.

Об'єктом дослідження є процес віддаленого адміністрування комп'ютера у мережі для організації техпідтримки.

Предметом дослідження є методи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки.

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

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

- Удосконалено метод віддаленого адміністрування комп'ютера у мережі для організації техпідтримки.
- Розроблено вітчизняний продукт віддаленого адміністрування комп'ютера у мережі для організації техпідтримки, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі віддаленого адміністрування комп'ютера у мережі для організації техпідтримки.

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

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

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 статті збірника праць молодих науковців ЦНТУ, випуск №14.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

КБПЗ – 2023

					VKPM-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Віддалене керування – це передача керування Інтернет віддаленим робочим столом, менеджменту комп'ютера або мережі комп'ютерів, а також інших робочих процесів за межі організації або домашньої мережі для спільної роботи. Віддалена робота з контролю при цьому виконується професіоналами в області ІТ-технологій і обслуговування комп'ютерів і обслуговування комп'ютерів.

Віддалене керування здійснюється за допомогою локальної або Інтернет-мережій програми для вебінарів. Тобто, компанія, що надає дані послуги, має доступ до необхідних мереж, комп'ютерам і системам через локальну мережу або мережу Інтернет і може стежити за станом певних сегментів керованої ІТ-мережі, вживати заходів по усуненню неполадок, діагностувати стан системи.

Такого роду аутсорсинг може здійснюватися як стосовно приватної ІТ-системи, так і щодо мережі комп'ютерів великої компанії й так далі. При цьому, приміром, контроль над домашнім комп'ютером здійснюється набагато легше, ніж ціла мережа з декількох десятків комп'ютерів, або повністю автоматизована ІТ-система.

Виходячи з великого досвіду надання послуги віддаленого керування в нашій країні, можна зробити висновки, що даний сервіс необхідний у таких випадках:

- для контролю над домашнім комп'ютером через недостатню компетентність його власника;
- для керування локальною мережею офісних комп'ютерів, поза залежністю від їхнього числа;
- набагато простіше проходить робота віддаленим співробітником

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

- для контролю над різного роду ІТ-системами;
- для того, щоб знати як проходить автоматизація бізнесу, контролю, обліку та ін.

Таким чином, віддалене керування має досить широке застосування й може бути використане стосовно самим різного за структурою й складністю ІТ-системам і програмам підтримки бізнесу.

1.2 Область застосування

Служба технічної підтримки та техпідтримка (Technical support, Helpdesk, Service desk) – сервісна структура, призначена для вирішення проблем користувачів з комп'ютерами і оргтехнікою (як апаратним, так і програмним забезпеченням). Важлива функціональна складова ІТІЛ (бібліотека інфраструктури інформаційних технологій), що дозволяє виявити проблемні ділянки інфраструктури ІТ, оцінити ефективність роботи підрозділу ІТ.

Служба технічної підтримки на кожному підприємстві може бути побудована різноманітними способами (мається на увазі реалізації процесів підтримки). Існує кілька моделей служби підтримки, наприклад: централізована, локальна, віртуальна – з єдиним телефонним центром і т. д. Служба технічної підтримки може бути організована як в цілях обслужити зовнішніх клієнтів (аутсорсинг обслуговування комп'ютерів і т. п.), так і внутрішніх (підрозділ ІТ-департаменту на великих підприємствах).

В описі концепції ІТІЛ, побудованої на процесному підході, Service Desk є єдиним описаним функціональним підрозділом. Цей виняток зроблено через велику важливість підрозділи техпідтримки при впровадженні практичному використанні сучасних ІТ-підходів та методик.

Правильно організована техпідтримка (Service Desk) завжди починається з реєстрації всіх звернень кінцевих користувачів, служить єдиною точкою для спілкування користувача з ІТ-службою. Найпопулярніші рішення з практичної

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

організації техпідтримки часто будуються на базі Call-center (іноді навіть користувачі їх ототожнюють). Він є початковою точкою контактів кінцевих користувачів зі службою технічної підтримки і служить джерелом інформації про їх фактичної задоволеності рівнем сервісу, що доповнює інформацію про технічні параметри якості обслуговування компанії-клієнта (зовнішнього або внутрішнього).

На великих підприємствах або у великих компаніях-аутсорсерах служба технічної підтримки часто організована за наступним багаторівневому принципі:

Користувач – звертається з питанням в службу підтримки по телефону або за допомогою електронної заявки.

Оператор (1-а лінія підтримки, Call-center) – реєструє звернення, при можливості допомагає користувачеві самостійно, або ескалює (передає і контролює виконання) заявку на другу лінію підтримки.

Друга лінія підтримки – отримує заявки від першої лінії, працює за ним, при необхідності залучаючи до вирішення проблеми фахівців із суміжних відділів (системні адміністратори, підтримка POS-терміналів, підтримка спеціального ПЗ, підтримка спеціального обладнання (ділінг) і т. д.)

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

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

Програми віддаленого керування комп'ютером дозволяють дистанційно управляти іншим комп'ютером через Інтернет або в локальній мережі. Це зручно, коли необхідно допомогти не дуже досвідченому користувачеві, наприклад, родичеві або другу, що погано розбирається в комп'ютері що-небудь зробити на ньому, не сходячи зі свого затишного крісла й не витрачаючи при цьому свої нерви й час на телефонні розмови. Такі програми ще зручно використовувати для віддаленої роботи, наприклад, з будинку для підключення в офіс і навпаки – для доступу до свого домашнього ПК, для системного адміністрування цілого парку комп'ютерів і серверів.

Проведемо порівняльний аналіз програм віддаленого керування, виділимо їхні переваги й недоліки.

TeamViewer

Одна із самих популярних програм для віддаленого доступу, її можна швидко скачати й встановити або відразу запусити, без установки, із цим зможе впоратися навіть не дуже досвідчений користувач. При запуску програма відображає вікно з ID і паролем для доступу до даного комп'ютера, а також TeamViewer дозволяє підключитися до іншого комп'ютера задавши його ID і пароль.

Переваги

У програмі доступно кілька основних режимів роботи це віддалене керування, передача файлів, чат, демонстрація свого робочого стола. Програма дозволяє налаштувати цілодобовий доступ до комп'ютера, це буде зручно для

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

системного адміністрування. Швидкість роботи цілком гідна, є версії для всіх мобільних платформ, для різних операційних систем, що дуже радує. Простий і цілком зрозумілий інтерфейс плюс ряд додаткових утиліт для розширення функціонала програми, будуть корисні для служб віддаленої підтримки.

Недоліки

Хоч програма і є безкоштовною, але тільки для не комерційного використання, а також при роботі з нею більше 5 хвилин виникає ряд труднощів, наприклад TV може заблокувати сеанс віддаленого підключення, розпізнавши його як комерційне використання. Для цілодобового віддаленого доступу або адміністрування декількох комп'ютерів, комп'ютерної мережі, прийдеться платити за додаткові модулі програми. Вартість програми висока.

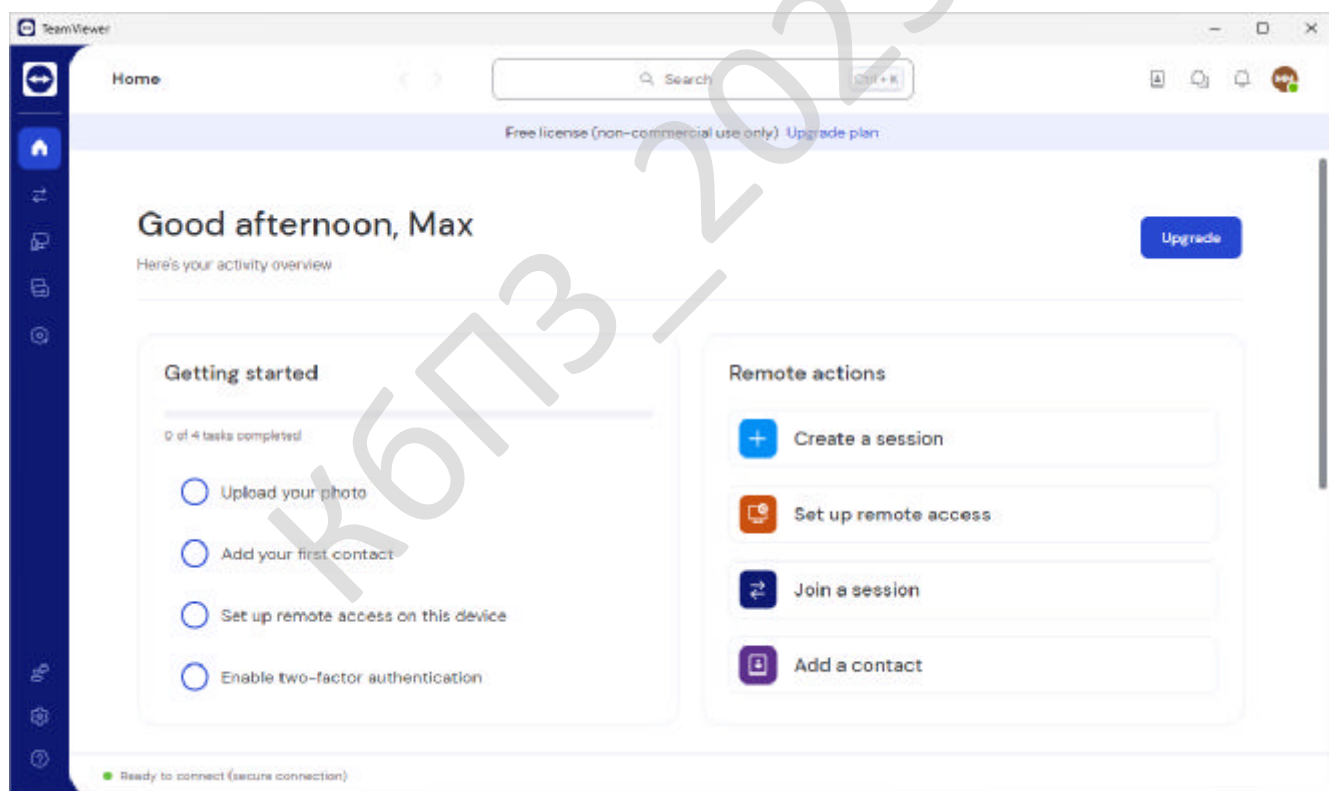


Рисунок 2.1 – Інтерфейс користувача TeamViewer

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

Підсумок

Дана програма ідеально підійде для разового віддаленого підключення або використання її нетривалі періоди часу. Зручно використовувати з мобільних платформ, але не адмініструвати велика кількість комп'ютерів. За додаткові модулі прийдеться доплачувати.

LiteManager

Проста, але досить таки потужна по можливостях програма, складається із двох частин, перша це Server який потрібно встановити або запустити на віддаленому комп'ютері й Viewer, що дозволяє управляти іншим комп'ютером.

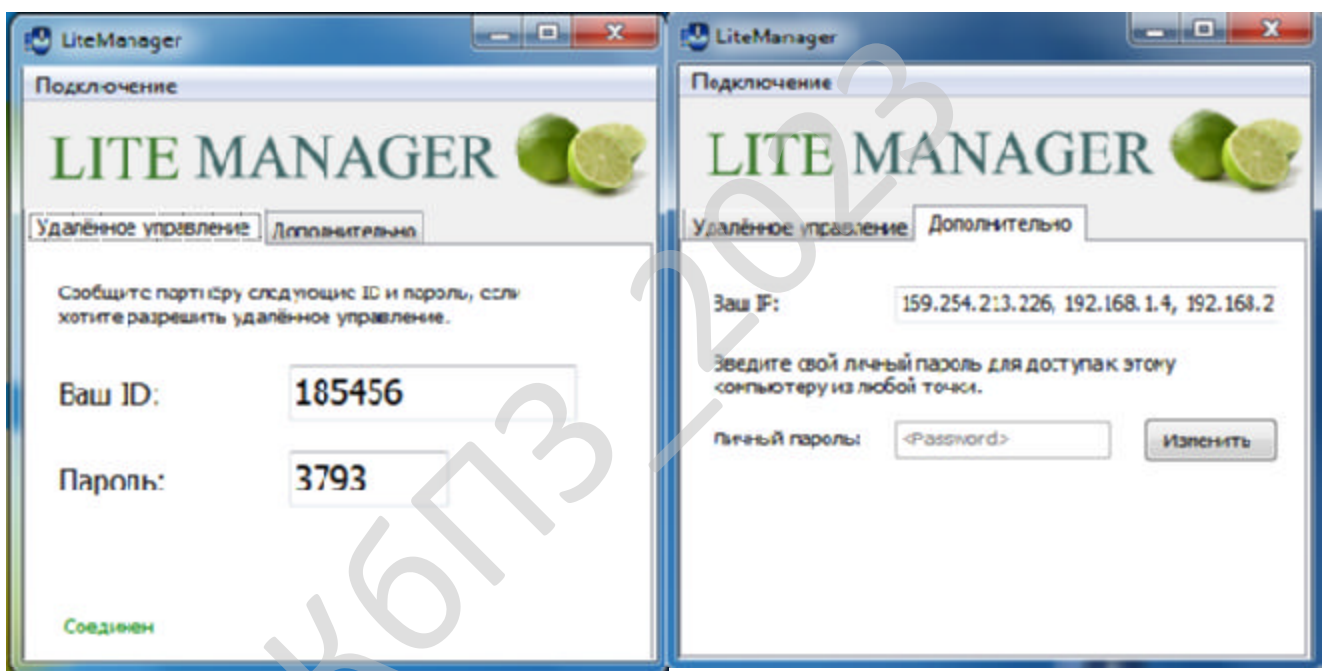


Рисунок 2.2 – Інтерфейс користувача LiteManager

Для роботи програма вимагає трохи більше навичок і досвіду від керуючого, хоча робота сервером навіть простіше ніж в TeamViewer, сервер можна один раз установити й більше не яких дій від користувача не потрібно, ID буде завжди постійний, його навіть можна задати самому в ручну, що дуже зручно для запам'ятовування. Версія LiteManager Free є безкоштовною для особистого й комерційного використання.

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

Переваги

У програмі крім основних режимів віддаленого доступу: віддаленого керування, передачі файлів, чату, диспетчера завдань, редактори реєстру, є й унікальні функції, наприклад: інвентаризація, запис екрана, віддалена установка. Програма безкоштовна для використання на 30-ти комп'ютерах, її можна використовувати для цілодобового доступу без яких або додаткових модулів. Відсутні які-небудь обмеження за часом роботи. Є можливість налаштування свого власного ID сервера для налаштування корпоративної служби підтримки. У програмі немає яких-небудь обмежень за часом роботи й блокувань.

Недоліки

Не вистачає клієнта під мобільні платформи або інші системи, є обмеження на 30 комп'ютерів у безкоштовній версії, для адміністрування більшої кількості необхідно придбати ліцензію. Деякі, специфічні режими роботи доступні тільки в Pro версії.

Підсумок

Програма LiteManager підійде для надання віддаленої підтримки, для адміністрування декількох десятків комп'ютерів абсолютно безкоштовно, для налаштування власної служби віддаленої підтримки. Вартість програми найнижча у своєму сегменті й ліцензія не обмежена за часом.

Ammy admin

Програма в основному аналогічна TeamViewer, але більше простий варіант. Присутні тільки основні режими роботи – перегляд і керування, передача файлів, чат. Програма може працювати без установки, безкоштовна для некомерційного використання.

Переваги

Проста й легка програма, можна працювати як в Інтернеті, так і в локальній мережі, має мінімальні налаштування й не вимагає якихось особливих умінь і навичок. По порівнянню з TeamViewer більше м'яка ліцензійна політика.

					VKPM-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

Недоліки

Мінімум функцій для віддаленого керування, адмініструвати великий парк комп'ютерів буде складно, при довгому використанні, більше 15 годин на місяць, сеанс роботи може бути обмежений або заблокований, платна для комерційного використання.

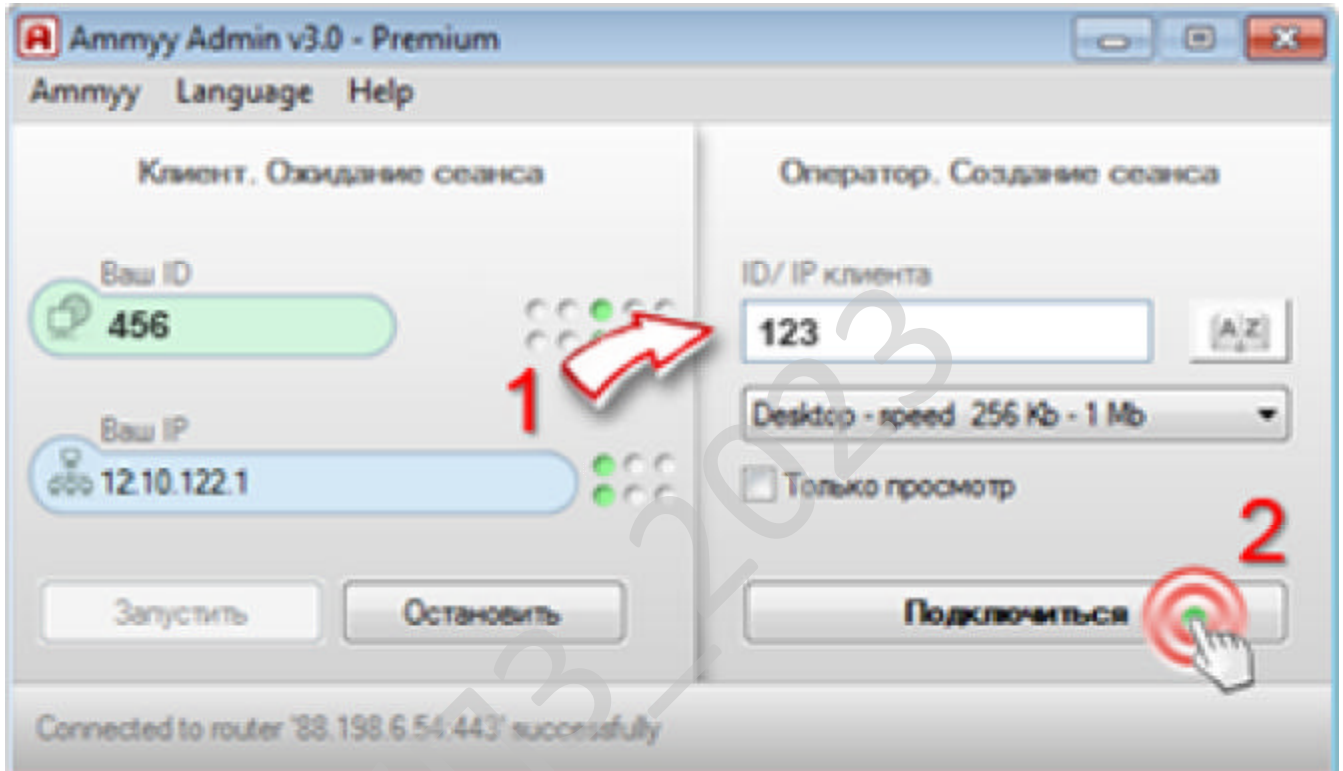


Рисунок 2.3 – Інтерфейс користувача Ammy admin

Підсумок

Дана програма більше підійде для разового підключення до комп'ютера й не сильно складних маніпуляціях, наприклад як надання допомоги не досвідченому користувачеві в налаштуванні комп'ютера.

RAdmin

Одна з перших програм віддаленого керування й відома у своєму колі, не міг її не згадати, більше призначена для системного адміністрування, основний акцент зроблений на безпеці. Програма складається із двох: компонент сервера й

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

клієнта. Вимагає установки, не досвідченому користувачеві буде не просто з нею розібратися, програма призначена в основному для роботи з IP адреси, що не зовсім зручно для надання тех. підтримки через Інтернет. Програма платна, але має безкоштовний тестовий період.

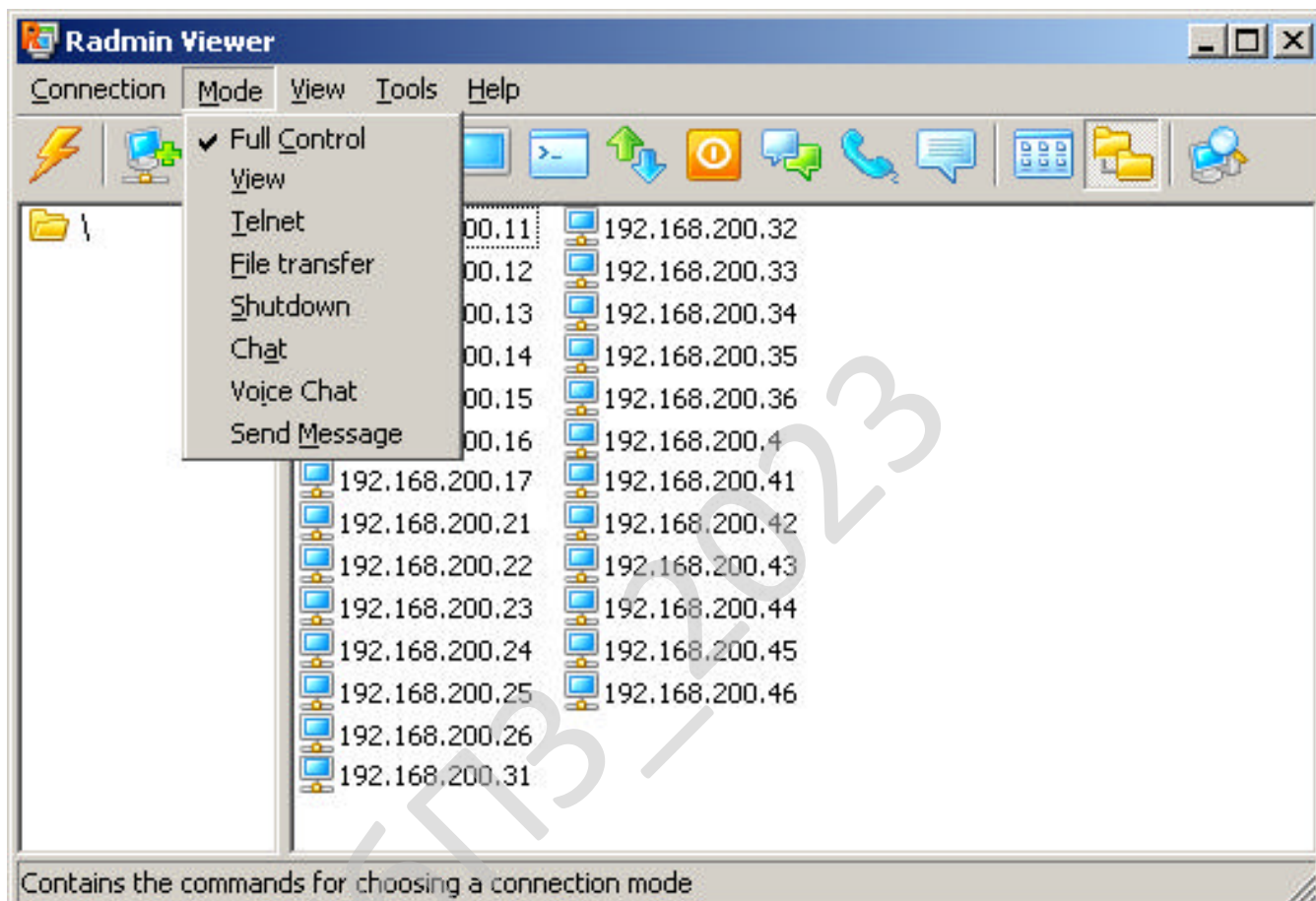


Рисунок 2.4 – Інтерфейс користувача RAdmin

Переваги

У програми висока швидкість роботи, особливо в гарній мережі, завдяки відеодрайверу захвата робочого стола, підвищеною надійністю й безпекою. Убудовано технологію Intel AMT, що дозволяє підключатися до BIOS віддаленого комп'ютера й набудувувати його. Реалізовані тільки основні режими роботи віддалене керування, передача файлів, чат і т.д.

Недоліки

Майже немає можливості для роботи без IP адреси, тобто з'єднуватися по ID. Відсутній клієнт для мобільних систем. Немає безкоштовної версії, тільки тестовий період 30 днів. Для роботи із програмою необхідні навички досвідченого користувача. При підключенні відео драйвер може відключати графічну оболонку Aero, іноді мигає екран.

Підсумок

Програма більше підійде для системних адміністраторів для адміністрування комп'ютерів і серверів у локальній мережі. Для роботи через Інтернет, можливо, прийдеться налаштувати VPN тунель.

Висновок

Є ще багато аналогічних програм для віддаленого доступу, це більше нові (Airoadmin, Supremo), дещо застарілі (VNC, Dameware, PCAnywhere) і інші більше дорогі гарні, але виконуючі ті ж самі основні функції. Якщо вирішите придбати ліцензію, зверніть увагу, що деякі програми ліцензуються на тимчасове користування на рік, після чого потрібно платити ще й ще.

Думаю, вибрати підходящу програму для себе ви зможете самі, або найкраще використовувати кілька рішень у комплексі.

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. 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.

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

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 підтримуватиме 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.

– Вбудований Fmxlinux.

– Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.

Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

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 звільнюються 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.

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.

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

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

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

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 віддаленого адміністрування комп'ютера у мережі для організації техпідтримки.

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

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

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

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

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

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-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Переваги віддаленої підтримки

Оперативність

Фахівцеві служби техпідтримки користувачів не потрібно виїжджати до робочого місця користувача (в офіс або додому), щоб вирішити проблему з комп'ютером, а досить запустити віддалений робочий стіл.

Широкі можливості для налаштування програмного забезпечення, устаткування й бізнес програмного забезпечення

С допомогою віддаленого доступу можна вирішувати проблеми, пов'язані із програмним забезпеченням, уникаючи відряджень: установка й налаштування програм, налаштування операційної системи, а іноді навіть і з устаткуванням (наприклад, проблеми з підключенням) – у цьому випадку теж можна допомогти віддалена підтримка, якщо направляти дії клієнта (у тому числі й за допомогою голосових повідомлень).

Економія часу й грошей

Вдаючись до допомоги віддаленої підтримки, вдається заощадити значну суму коштів і часу, як з боку служби техпідтримки, так і для користувача завдяки так званим відрядженням онлайн.

Невимогливість до каналу зв'язку й апаратному забезпеченню

Сучасне програмне забезпечення оптимізоване для каналів зв'язку з невисокою пропускнуою здатністю – можна віддалено працювати не тільки через ADSL, але й по GPRS і навіть по dial-up. Також віддалена підтримка не вимагає потужного комп'ютера – підійде й звичайна офісна робоча конячка.

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

Високий рівень безпеки

Всі передані дані (зображення на екрані, текстові й голосові повідомлення) піддаються шифруванню за допомогою стійких криптоалгоритмів. Є можливість парольного захисту й фільтрації доступу по IP-адресах (тобто доступ можна дозволити тільки із заданих вами комп'ютерів і підмереж).

Розмаїтість способів як управляти віддаленим комп'ютером

Програмне забезпечення для віддаленої підтримки може працювати в декількох режимах – керування комп'ютером віддалено, асистування, перегляд робочого стола, копіювання файлів на комп'ютер користувача й назад, адміністрування.

Нові можливості для навчання

Використовуючи віддалену підтримку в режимі асистування, можна організувати дистанційне навчання клієнтів або студентів: проводити навчальні онлайн семінари, віддалені курси семінари й т.п.

Віддалене керування комп'ютером через Інтернет

Віддалене керування комп'ютером через Інтернет – це можливість використовувати комп'ютер з будь-якої точки світу так, ніби ви сиділи безпосередньо за ним. Скрізь, де є Інтернет, у вас з'являється можливість підключатися до робочого або домашнього комп'ютера й використовувати його можливості на сто відсотків для рішення різних завдань.

Віддалене керування комп'ютером через Інтернет застосовується в наступних випадках:

– Для одержання доступу до офісного комп'ютера або файл-серверу з будинку на вихідних, з відрядження або ж з іншого офісу віддалене керування комп'ютером. Ви зможете користуватися повним набором ліцензійного ПЗ на своєму робочому комп'ютері з будь-якої точки, одержите доступ до всіх своїх документів і зможете вести офісну роботу навіть із кафі. Подібний вид віддаленого керування комп'ютером через Інтернет активно використовується часто подорожуючими бізнесменами по усьому світі. Просто не виключайте офісний комп'ютер – і він буде «з вами» скрізь, де у вас є вихід у мережу.

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

- Для керування критичними комп'ютерами, що працюють у складних мережах, наприклад, серверами. Віддалене керування комп'ютером через Інтернет дозволяє системному адміністраторові постійно мати доступ до ресурсу, проводити онлайн наради на збої в роботі якого повинна впливати негайна реакція.

- Для роботи з віддаленими комп'ютерами й мережами через віддалений робочий стіл. Завдяки можливості керування через Мережу, технічні фахівці одержують можливість обслуговувати будь-які системи, що перебувають у самих різних точках планети. Один системний адміністратор, приміром, може набувати комп'ютери відразу в декількох містах: це дуже зручно для компаній, що мають більшу мережу філій або складну інформаційну інфраструктуру.

- Для проведення вебінарів, тобто дистанційного навчання в ході онлайн семінари. Можливість безпосередньо працювати на комп'ютері викладача або ж можливість викладача бачити всі ваші дії й допомагати вам – це серйозна підмога в утворенні особливо для викладання англійський онлайн.

- Для часто подорожуючих: за допомогою будь-якого нетбука з виходом в Інтернет ви можете здійснювати віддалене керування домашнім комп'ютером, перевіряти пошту й планувати завдання, проводити навчання онлайн.

Для організації віддалене керування комп'ютером через Інтернет необхідно:

- По-перше, установка спеціальної програми на комп'ютер, що буде управлятися. Ця програма дозволяє захистити ваші дані, тобто забезпечити конфіденційність обміну інформацією, а також правильність і надійність авторизації. Тільки ви зможете управляти своїм комп'ютером – і будете при цьому впевнені в тому, що всі передані дані надійно захищені для навчання онлайн.

- По-друге, залишити керований комп'ютер включеним і підключеним до мережі для керування комп'ютером віддалено. Це цілком нормальний стан для

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

сервера, але психологічно незвичне для домашнього комп'ютера. Не турбуйтеся, з ним усе буде в порядку. Крім того, ви зможете самі виключити його в потрібний час, використовуючи всі те ж віддалене керування remote dekstop.

– По-третє, використання програми або мережного сервісу на тім комп'ютері, з якого ви збираєтеся здійснювати доступ, проводити безкоштовні презентації. Технічно можливе підключення як з будь-якого терміналу Інтернет-кафе, так і з вашого нетбука або навіть мобільного телефону. Все залежить тільки від продажу пропонованого сервісу.

3.2 Розробка структурної схеми

Структурна схема розробленого програмного забезпечення системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки складається з наступних шести компонентів:

- Головний сервер.
- Сховище даних.
- SQL-сервер і база даних.
- Блок системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки.
- Клієнтський реєстратор.
- Центр керування.

Головний сервер

Програмне забезпечення основного сервера, що звичайно розташовується на тій же машині, що й SQL-сервер бази даних, забезпечує такі послуги, як відновлення програмного забезпечення й перевірку ліцензій для всіх комп'ютерів, зконфігурованих під додаток «Клієнтський реєстратор».

Сховище даних

Всі додатки «Клієнтський реєстратор» передають свою інформацію в сховище «Сховище даних» через попередньо встановлені інтервали часу. Ця

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

інформація містить дані про подію, образи екранів і вкладення в електронну пошту. Потім «Сховище даних» пересилає ці дані в базу даних на SQL-сервері й зберігає образи екранів і вкладення електронної пошти в мережному каталозі, доступному через додаток «Блок системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки».

SQL-сервер і база даних

SQL-сервер і база даних зберігають всі записані події, зібрані за період часу на контрольованих комп'ютерах. Керування базою даних, включаючи резервне копіювання й архівування, виробляється через додаток «Блок системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки».

Блок системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки

Додаток «Блок системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки» є центральним пунктом для моніторингу дій на віддаленому комп'ютері у розроблене програмне забезпечення системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки. Додаток «Блок системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки» направляє запити до SQL-сервера на одержання даних про подію й витягає образи екрана й поштові вкладення з мережного каталогу. «Блок системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки» є додатком Windows на основі графічного інтерфейсу .NET. Є можливість установити в мережі більше одного додатка «Блок системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки». Також можливо зконфігурувати кілька баз даних, до кожної з яких буде доступ за допомогою «Блок системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки».

Основний сервер, сховище «Сховище даних» і SQL-сервер з базою даних можуть перебувати на одному комп'ютері рівня сервера або можуть бути розподілені між декількома машинами для поліпшення продуктивності. Додаток

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

«Блок системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки» і «Центр керування» можуть перебувати на одній машині, але дуже часто ці компоненти ізольовані друг від друга по міркуваннях безпеки. Наприклад, особа, що відповідає за установку клієнтських компонентів, може бути не вповноважене для перегляду зібраних даних. В організації можуть бути розгорнуті кілька копій додатка «Блок системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки». Наприклад, можна надати копію додатка менеджерів по кадрам або лінійному керівникові, щоб він міг спостерігати за співробітниками для реалізації системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки. Перегляд з кожного екземпляра «Блок системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки» може бути обмежений для особи, що виконує аналіз даних, доступом тільки до інформації, що має відношення до системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки.

Запис подій

У розробленому програмному забезпеченні системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки реалізована передова технологія моніторингу комп'ютерів, що автоматично записує всі дії, включаючи що відправляються й прийняті повідомлення електронної пошти, спілкування в чатах і системах миттєвого обміну повідомленнями, відвідувані веб-сайти, набрані на клавіатурі дані, передані / надруковані / збережені файли й багато чого іншого.

Розроблене програмне забезпечення системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки «Реєстратор» містить кілька потужних засобів запису:

- Реєстратор електронної пошти.
- Реєстратор відвідування веб-сайтів.
- Використання пошуку он-лайн.

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

- Реєстратор сесій чату / миттєвих повідомлень.
- Реєстратор клавіатурного уведення.
- Реєстратор програмної активності.
- Реєстратор мережної активності.
- Реєстратор передачі файлів.
- Реєстратор роботи з документами.
- Реєстратор образів екрана (відеозапис).
- Запис активності / неактивності користувача.
- Визначення й виявлення ключових слів.

Всі ці інструменти ведуть запис одночасно, сховано, під захистом потрібного рівня безпеки. Додаток «Реєстратор» добре конфігурується й може бути настроєне для запису тільки подій, що цікавлять.

На додаток до моніторингу й ведення запису розроблене програмне забезпечення системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки має розвитку систему визначення й виявлення ключових слів, що буде негайно сповіщати про кожний випадок, коли користувач контрольованого ПК відхилиться від припустимого використання ПК або Інтернет.

Реєстратор електронної пошти

Розроблене програмне забезпечення системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки буде відзначати зловживання за допомогою передових засобів запису електронної пошти, які покликані забезпечити повну реєстрацію всіх електронних повідомлень, прийнятих і відправлених з комп'ютерів організації. На додаток до реєстрації електронної пошти, що проходить через стандартні служби MS Exchange і Outlook, розроблене програмне забезпечення системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки включає засобу захвату електронної пошти, що відправляється через веб-служби. Сюди входять поштові служби, такі як Hotmail, Yahoo Mail і AOL Internet Email.

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

Щораз при одержанні або відправленні електронного повідомлення, розроблене програмне забезпечення системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки створює схований дублікат (включаючи вкладення в електронне повідомлення) і зберігає його для наступного перегляду. Розроблене програмне забезпечення системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки також зберігає копії вилучених листів.

Вкладення в перехоплені електронні листи надійно передаються в сховище «Сховище даних» із застосуванням шифрування даних.

Якщо хочете обмежити кількість записуваних електронних листів, цей інструмент має засобу фільтрації, так що листи будуть перехоплюватися при відправленні або одержанні тільки при наявності певного шаблону, виявленого в адресах, заголовку або тілі листа. Інші засоби керування дозволяють відфільтрувати листи на основі використовуваного формату або при виявленні вкладень. На додаток до цього є укрупнене налаштування для фільтра, засноване на характеристиках вкладення, наприклад, його розмірі. Якщо хочете перехоплювати листи тільки з певного джерела, можна відфільтрувати листи, отримані з певних типів облікових записів, таких як MS Outlook, Hotmail, AOL, MS Exchange або IMAP.

Можливості реєстратора електронної пошти:

- SMTP / POP, IMAP, Web.
- Exchange / MAPI.
- Exchange сканує тільки папку вхідних на наявність існуючих нових листів.
- Електронна пошта AOL.
- Перевірка дублікатів листів.
- Сканування листів по ключових словах.
- Захват вкладень.

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

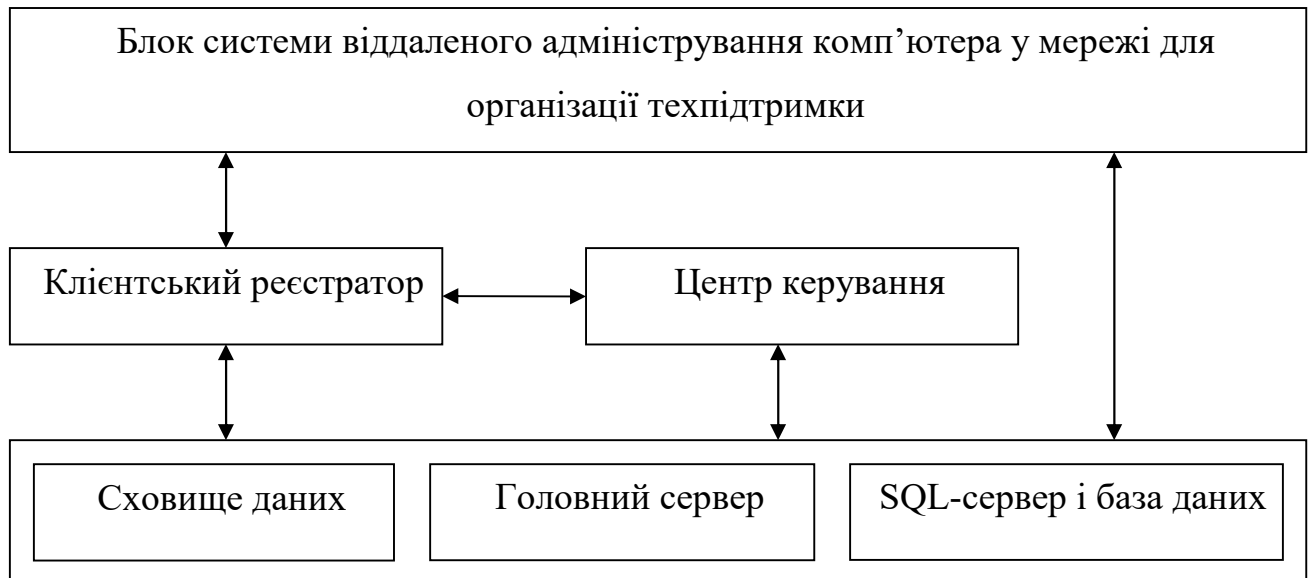


Рисунок 3.1 – Структурна схема системи

Реєстратор відвідування веб-сайтів

Засоби реєстрації відвідувань веб-сайтів у пакеті розроблене програмне забезпечення системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки будуть постійно відслідковувати кожну веб-сторінку, що відвідали користувачі комп'ютерів у Вашій мережі, і збережуть запис про таких URL і доменах.

Розроблене програмне забезпечення системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки також буде відслідковувати кожний файл і програму, завантажені з мережі Інтернет (такі як файлообмінні сервіси, сайти для завантаження ПЗ, порнографічні зображення й т.п. ...)

Розроблене програмне забезпечення системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки більше розвинена, чим функція журналу веб-браузера, і забезпечує важливу інформацію, включаючи наступне: Час останнього відвідування, тривалість перебування на веб-сайті, активний час на веб-сайті, загальна кількість відвідувань веб-сайту, назва сторінки й інші дані.

На додаток до цього можна використовувати центр керування розроблене програмне забезпечення системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки для створення списку дозволених веб-сайтів, які можуть відвідувати співробітники організації. Будь-які веб-сайти, не включені в цей перелік, будуть заблоковані. І навпаки, можна створити "чорний список" веб-сайтів, доступ до яких заборонений. Якщо веб-сайт не включений у цей список, на нього можна зайти. Ці політики можна налаштовувати для окремих співробітників.

За допомогою розвинених засобів реєстрації відвідування веб-сайтів розроблене програмне забезпечення системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки можна швидко одержати точну картину поведінки кожного окремого співробітника, що також використовується в додатку «Блок системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки» для формування даних про тенденції по всій організації.

Можливості реєстратора відвідування веб-сайтів:

- Сканування адрес веб-сайтів по ключових словах.
- Сканування сторінок веб-сайтів по ключових словах.
- Захват браузерів Mozilla (наприклад, Firefox).
- Захват AOL Security Edition.

Запис пошукових запитів

Розроблене програмне забезпечення системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки записує запити до пошукових машин з будь-якого веб-сайту, де використовуються відомі технології пошуку. Розроблене програмне забезпечення системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки переглядає стандартні пошукові технології, захоплює пошукові фрази, введені користувачем, посилання URL результатів пошуку й іншу інформацію. Використовуючи додаток «Блок системи віддаленого адміністрування комп'ютера у мережі для організації

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

техпідтримки» можна довідатися, чи дійсно співробітник відвідував веб-сайт, отриманий у результаті пошукового запиту. Ця інформація дозволяє зрозуміти, чому співробітник був на цьому сайті. Наприклад, співробітник шукав сайти по працевлаштуванню з метою професійного росту або щоб знайти нову роботу.

При пошуку он-лайн захоплюються такі сайти: google.com, msn.com, yahoo.com. ask.com, altavista.com, gigablast.com. alltheweb.com, go.com, live.com, teoma.com. і інші....

Реєстратор сесій чату / миттєвих повідомлень

Розроблене програмне забезпечення системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки забезпечує автоматичну реєстрацію всіх сесій чату й миттєвих повідомлень, щоб надати повну інформацію про кожний комп'ютерний обіг, що мало місце у Вашій організації або з Вашими клієнтами. Розроблене програмне забезпечення системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки записує імена всіх учасників сесій чату / миттєвих повідомлень і зберігає повну копію спілкування. Ця інформація записується для AOL, MSN, ICQ і інших різних програм чату / миттєвих повідомлень.

Можливості реєстратора сесій чату / миттєвих повідомлень:

- Сканування чату й миттєвих повідомлень по ключових словах.
- Захват IRC, MSN, AIM / ICQ, AOL, Yahoo.
- Захват XMPP, OSCAR80, MSN Exchange.
- Захват миттєвих повідомлень через веб-форми.

Реєстратор клавіатурного уведення

Розроблене програмне забезпечення системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки містить удосконалений реєстратор клавіатурного уведення, що зберігає уведені із клавіатури дані, організовані по додатках, по даті й часу.

За допомогою розробленого програмного забезпечення системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

будете знати, які дані були набрані, коли й де вони були уведені, і також будете знати, хто їх увів, тому що розроблене програмне забезпечення системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки зберігає запис про те, під яким ім'ям користувач був зареєстрований на комп'ютері.

Реєстратор клавіатурного уведення розроблене програмне забезпечення системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки буде документувати будь-яке натискання клавіші на клавіатурі ПК. Розроблене програмне забезпечення системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки не тільки захоплює стандартні алфавітно-цифрові клавіші, але також записує "сховані" символи й комбінації клавіш, таких як Shift і Ctrl, а також ті символні послідовності, які звичайно не відображаються на екрані, наприклад, при уведенні пароля відображаються тільки зірочки.

Можливості реєстратора клавіатурного уведення:

- Сканування уведених рядків по ключових словах.
- Захват символної інформації.

Реєстратор програмної активності

До складу розробленого програмного забезпечення системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки входять засоби моніторингу, які забезпечують чіткий і лаконічний огляд програм, використовуваних на офісних ПК. Розроблене програмне забезпечення системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки буде записувати дату й час запуску програми й загальний час активності цієї програми. Також записується специфічна для програми інформація, включаючи назву її вікна.

Можливості реєстратора програмної активності:

- Налаштування тайм-ауту неактивності.

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

Основне питання полягає в тому, чи передають співробітники конфіденційну інформацію компанії через Інтернет?

Інша проблема полягає в тому, чи піддають співробітники свою організацію правовому ризику, завантажуючи захищені авторським правом матеріали, такі як музика, фільми й комерційне ПЗ? Крім того, є потенційна погроза, коли співробітники завантажують порнографічні зображення й залишають їх на комп'ютерах компанії, де вони можуть потрапити на очі іншим співробітникам, які вправі порахувати такі матеріали дуже образливими.

І нарешті, існує великий ризик того, що щораз при завантаженні файлу може бути встановлене шпигунське або рекламне ПЗ, особливо при використанні сервісів однорангових мереж. (Корпорація Dell Computer називає шпигунське ПЗ самим великим джерелом проблем, що вимагають технічної підтримки, оскільки це ПЗ перенаправляє веб-браузери й в остаточному підсумку сильно сповільнює роботу комп'ютерів, і це не вважаючи потенційну погрозу безпеки. Найбільша частина шпигунського ПЗ попадає в комп'ютери шляхом завантаження через Інтернет).

Розроблене програмне забезпечення системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки записує всі файли, передані через Інтернет, відслідковуючи передачу файлів по протоколах HTTP, FTP і через однорангові мережні з'єднання. Можна переглянути, хто завантажував або передавав файли, який протокол передачі застосовувався, яка служба використовувалася, ім'я й тип переданого файлу (аудіо, зображення, відео, програмне забезпечення, документ).

Можливості реєстратора передачі файлів:

– Захват вивантаження за протоколом HTTP.

Реєстратор роботи з документами

Крадіжки інтелектуальної власності й витік конфіденційної інформації стають в останні роки наростаючою погрозою безпеки компаній, оскільки технології дають можливість співробітникам відносно просто копіювати файли

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

на гнучкі диски, диски Zip, компакт-диски, DVD, приводи USB, карти пам'яті або просто роздруковувати ці документи й виносити їх за межі приміщень компанії.

Програма містить засоби реєстрації роботи з документами, які дозволяють визначити той факт, що файли / документи могли покинути приміщення організації на знімних носіях, таких як гнучкі диски, zip-накопичувачі, CD, DVD або USB-накопичувачі й карти пам'яті. Засоби реєстрації роботи з документами також дозволяють відслідковувати печатка документів і хто копіював або друкував документи або файли, піддані потенційному ризику.

Можливості реєстратора роботи з документами:

- Включення спостереження при створенні нового файлу.
- Включення спостереження при записі в існуючий файл.
- Включення спостереження при видаленні / перейменуванні файлу.
- Включення спостереження, коли місцем призначення при записі є привод CD / локальний диск / мережний диск.
- Включення спостереження, коли місцем призначення є знімний носій.
- Включення спостереження для інших місць призначення.
- Підтримка фільтрації по ім'ю файлу.
- Спостереження за роздруківкою документів.
- Відстеження запису на CD / DVD Burning (IMAPI) під WinXP.

Реєстратор образів екранів

Представте відеокамеру спостереження, спрямовану на кожний ПК у мережі організації, що знімає всі дії користувача в Інтернет і звичайну роботу на ПК, і зберігає записаний фільм у безпечному місці для наступних переглядів. Це ідея, на якій побудована робота реєстратора образів екрана в розроблене програмне забезпечення системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки.

Розроблене програмне забезпечення системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки робить сотні знімків щогодини на кожному ПК у мережі, дуже схоже на роботу камери відеоспостереження.

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

Зроблені знімки відбивають зміст екрана комп'ютера з обраною частотою – аж до одного знімка в секунду.

Це значить, що Ви одержуєте докладний візуальний запис всіх спілкувань у чаті, миттєвих повідомлень, набраних і прочитаних електронних листів, всіх відвіданих веб-сайтів, всіх запущених програм, всіх набраних на клавіатурі рядків, – всіх дій на всіх комп'ютерах.

Запис образів екранів є самим потужним засобом візуального запису й перегляду всього, що відбувається на комп'ютері.

Можливості реєстратора образів екранів:

- Захват вторинних моніторів.
- Установка графічного формату для захоплених образів екранів.
- Установка частоти знімків.
- Налаштування тайм-ауту неактивності.

Запис активності / неактивності користувача

Це засіб реєстрації в розроблене програмне забезпечення системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки збирає інформацію про активність і неактивність співробітників. Такі події як реєстрація на комп'ютері, використання миші й клавіатури, відсутність активності на комп'ютері – все це записується. Події, які можуть потенційно спотворити цю інформацію, також записуються (як приклад – зміна системного часу).

Можливості реєстратора активності / неактивності користувача:

- Формат таблиця обліку робочого часу.
- Табелю обліку робочого часу користувача.
- Індикатор входу в систему й активності.
- Індикатор виходу із системи й неактивності.
- Індикатор зміни годин системи.
- Індикатор припинення запису.

Визначення й виявлення ключових слів

Визначення й виявлення ключових слів дозволяє задавати слова або фрази, які можуть сигналізувати про безпосередню погрозу схоронності й цілісності мережних систем, інтелектуальної власності, з боку співробітників або клієнтів. Ключові слова можуть також використовуватися для вказівки, що співробітник бере участь у діяльності, що не схвалюється компанією або є прямим порушенням політики прийнятного використання.

Створіть список "тривожних" слів або фраз, і розроблене програмне забезпечення системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки буде автоматично відслідковувати дані, посилання, що набираються на клавіатурі, URL, що входить і виходить електронну пошту, спілкування в чаті й миттєві повідомлення й відразу ж повідомляти про такі терміни.

Коли виявляється ключове слово або фраза, розроблене програмне забезпечення системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки негайно переходить у режим перехоплення. Частота зйомки образів екрана автоматично підвищується, щоб забезпечити всі подробиці загрозованої діяльності й захвату їх як докази. На додаток до цього розроблене програмне забезпечення системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки збирає інформацію щодо цієї діяльності й негайно передає повідомлення про тривожний випадок по електронній пошті зазначеному ІТ-адміністраторові або відповідному менеджеру.

Кожне тривожне повідомлення містить наступну інформацію:

- Виявлене ключове слово.
- Час виявлення ключового слова.
- Поточний користувач, зареєстрований в Windows.

Можливості системи визначення й виявлення ключових слів:

- Імпорт ключових слів.
- Експорт ключових слів.

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

- Установка дії по ключових словах (наприклад, відправлення електронної пошти).
- Установка регулярності дій по ключових словах.
- Установка типу сервера e-mail (прямий SMTP, перехідний SMTP, папка обміну).
- Захват образів екранів при повідомленні із заданими інтервалом і тривалістю.

Схований режим

Реєстратор розробленого програмного забезпечення системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки можна перевести в схований режим, якою забезпечує неможливість виявлення програми з користувачами. У схованому режимі розроблене програмне забезпечення системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки не буде видний користувачеві в системному меню завдань, диспетчері завдань або в меню установки / видалення програм панелі керування.

Опції й параметри конфігурації клієнта розробленого програмного забезпечення системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки

Додаток «Реєстратор» відповідає за реєстрацію всіх дій і запис образів екранів зі спостережуваних комп'ютерів і передачу цієї інформації в додаток «Сховище даних». Додаток «Реєстратор» добре конфігурується й може бути тонко настроєне для кожного екземпляра установки. Звичайно «Реєстратор» конфігурується за допомогою розробленого програмного забезпечення системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки «Центр керування» і потім установлюється на клієнтській машині.

Клієнтський реєстратор

Цей програмний компонент конфігурується й установлюється на кожний комп'ютер, що підлягає моніторингу й може працювати в “схованому режимі”, так що співробітники не будуть знати про спостереження за ними. Додаток

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

«Клієнтський реєстратор» добре конфігурується й може використовуватися для фільтрації й блокування специфічного трафіку, диспетчеризації доступу до мережі, генерації оповіщення по ключових словах і інших функціях.

Центр керування

Всі клієнтські додатки «Клієнтський реєстратор» конфігуруються й управляються додатком розробленого програмного забезпечення системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки «Центр керування», що використовується для установки клієнтських додатків на комп'ютери, розташовані у внутрішній мережі компанії. Додаток «Центр керування» також відповідає за конфігурування сховища даних «Сховище даних».

3.3 Розробка функціональної схеми

На рисунку 3.2 зображена функціональна схема системи. Нижче розглянемо її більш докладно. Функціональна схема складається з двох великих блоків:

- Блок визначення функцій моніторингу.
- Блок визначення об'єктів моніторингу.

Блок визначення функцій моніторингу складається з наступних блоків:

- Робота з ресурсами мережі.
- Робота з сесіями.
- Моніторинг трафіку.
- Робота з файлами.
- Монітор з'єднань.
- Статистика подій.
- Функції для роботи з мережею.

Розглянемо детальніше кожний з блоків.

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.

З рисунку видно, що блок визначення об'єктів моніторингу локальної мережі складається з трьох блоків:

- Моніторинг трафіку.
- Моніторинг обладнання.
- Моніторинг ресурсів.

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.

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

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.

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44



Рисунок 3.2 – Функціональна схема системи

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.

TCP – один з основних мережних протоколів Інтернету, призначений для управління передачею даних в мережах і підмережах TCP/IP.

UDP – один із протоколів в стеку TCP/IP. Від протоколу TCP він відрізняється тим, що працює без встановлення з'єднання. UDP – це один з найпростіших протоколів транспортного рівня моделі OSI, котрий виконує обмін дейтаграмами без підтвердження та гарантії доставки.

IP – найбільш широко розповсюджена реалізація ієрархічної схеми мережної адресації. Використовуваний в мережі Інтернет, протокол відповідає за адресацію пакетів, але не відповідає за встановлення з'єднань, не є надійним і дозволяє реалізувати тільки негарантовану доставку даних.

ICMP – мережний протокол, що входить в стек протоколів TCP/IP. В основному ICMP використовується для передачі повідомлень про помилки й інші виняткові ситуації, що виникли при передачі даних. Також на ICMP покладають деякі сервісні функції, зокрема на основі цього протоколу заснована дія таких загальновідомих утиліт як ping та traceroute.

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, IP-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.

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

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

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.
- Моніторинг трафіку.
- Підключені інтерфейси.

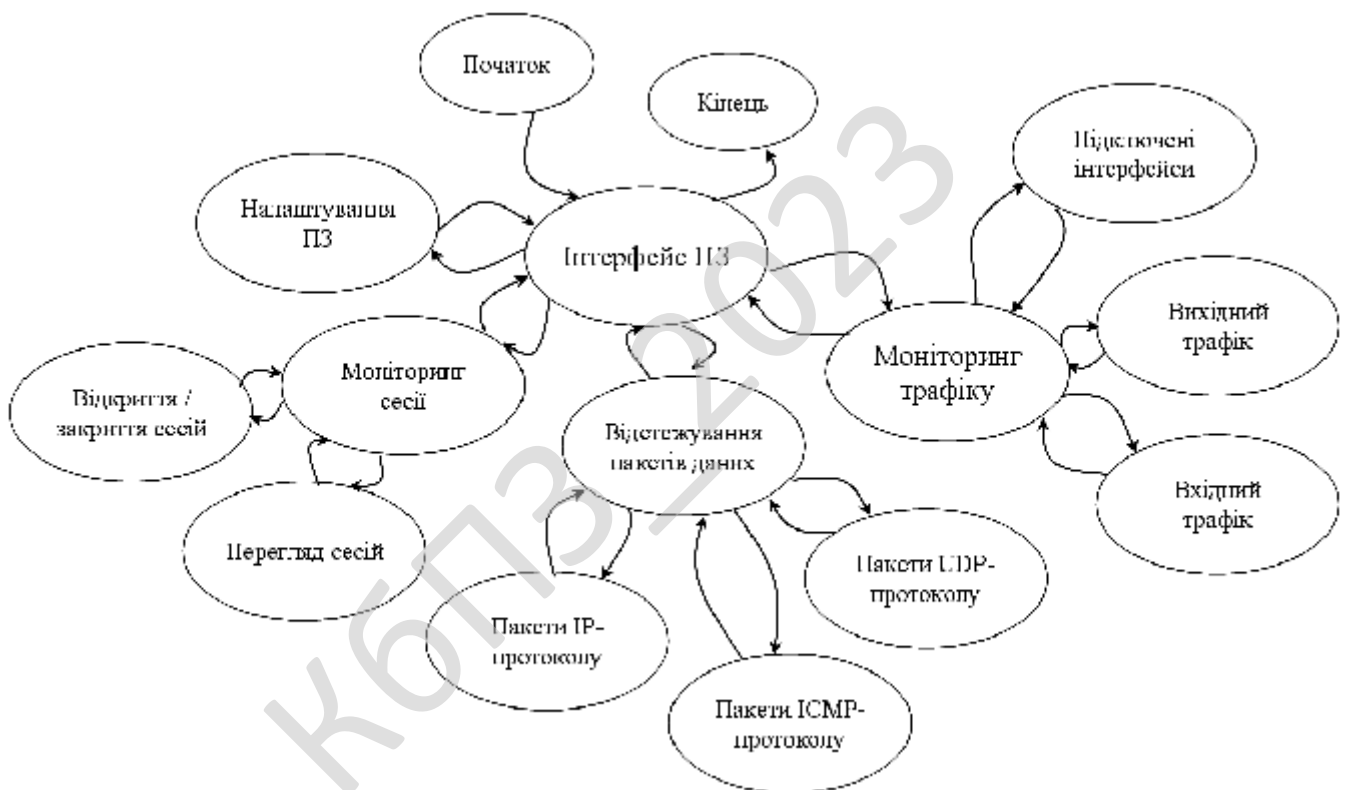


Рисунок 3.3 – Діаграма взаємодії процесів

- Вихідний трафік.
- Вхідний трафік.
- Налаштування ПЗ.
- Моніторинг сесій.
- Відкриття / закриття сесій.

- Перегляд сесій.
- Відстежування пакетів даних.
- Пакети UDP-протоколу.
- Пакети ICMP-протоколу.
- Пакети IP-протоколу.

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.

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

КБПЗ-2023

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

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.

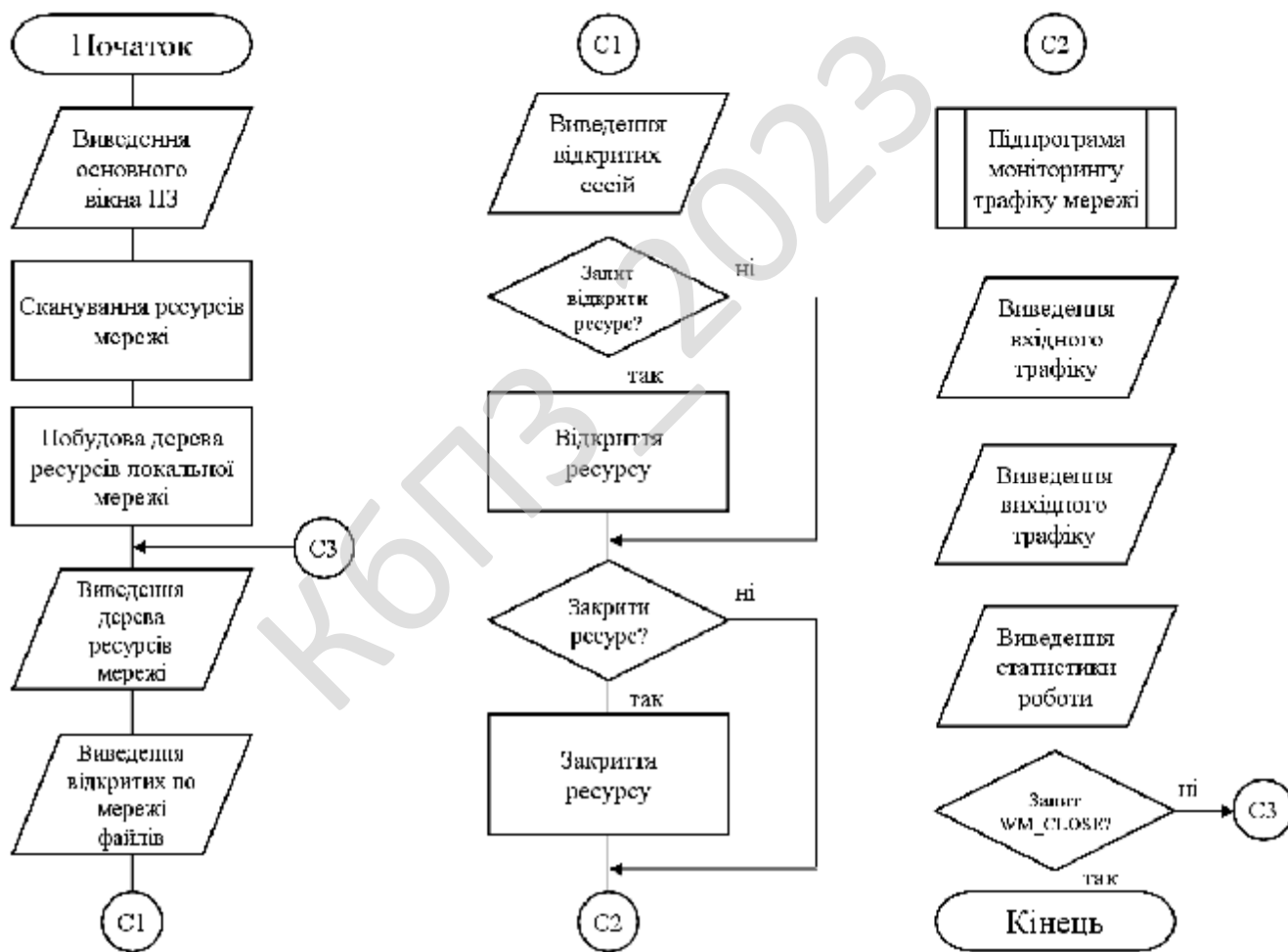


Рисунок 4.1 – Блок схема основної програми

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

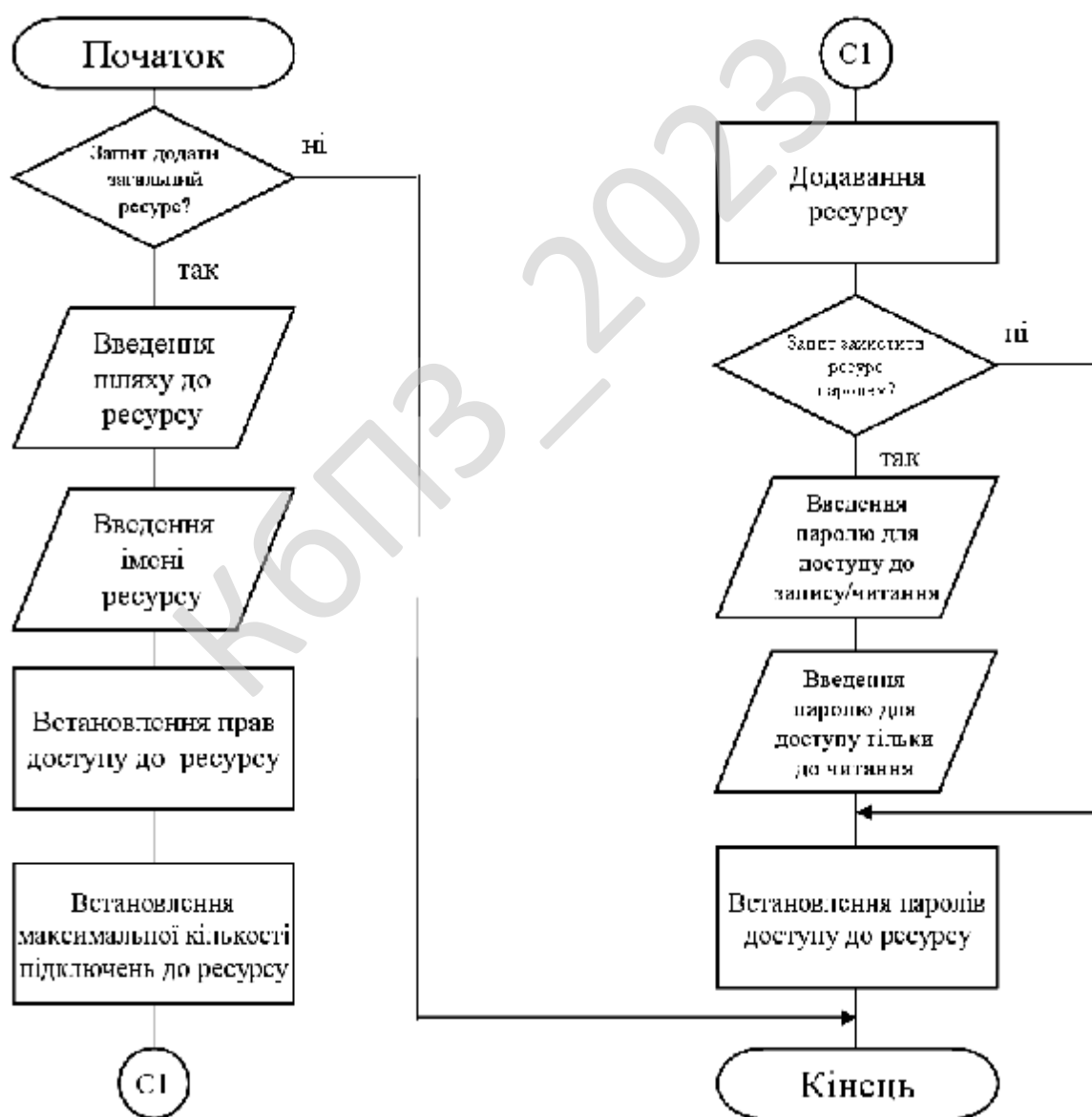


Рисунок 4.2 – Блок схема підпрограми

Опис алгоритмів функціонування системи

Розглянемо кроки реалізації системи. 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 системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки.

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.

Програма на початку роботи визначає під якою системою вона працює. Можливими є 2 варіанти:

- Система XP-типу.
- Система Windows 7 та вище.

Для отримання списку всіх відкритих загальних інструкцій в програмі використовується процедура TMainForm.btnGetS:

В цій процедурі використовуються наступні змінні:

```
var
  i:Integer;
  FLibHandle : THandle;
  ShareNT : PShareInfo2Array;
  entriesread,totalentries:DWORD;
  Share : array [0..512] of TShareInfo50;
  pcEntriesRead,pcTotalAvail:Word;
  OS: Boolean;
```

Тут PShareInfo2Array визначається як вказівник на масив вказівників на дані типу TShareInfo2, які являють собою структуру із наступними полями:

```
TShareInfo2 = packed record
  shi2_netname : PWChar;
  // мережне ім'я
```

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

btnGetSC, для якої наведемо основну частину виконуваного коду.

```
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
begin
Caption := string(SessionInfo502^[i].sesi502_cname);
SubItems.Add(SessionInfo502^[i].sesi502_username);
SubItems.Add(IntToStr(SessionInfo502^[i].sesi502_num_opens));
SubItems.Add(CardinalToTimeStr(SessionInfo502^[i].Sesi502_Time));
SubItems.Add(CardinalToTimeStr(SessionInfo502^[i].sesi502_idle_time));
end;
end;
```

Змінні, які використовуються в цій процедурі.

```
SessionInfo50: array [0..512] of TSessionInfo50;
SessionInfo502 : PSessionInfo502Array;
TotalEntries, EntriesReadNT: DWORD;
```

Тип PSessionInfo502Array визначається наступним чином.

```
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;
```

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54


```

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
begin
Caption := string(IntToStr(FileInfoNT^[i].fi3_id));
SubItems.Add(FileInfoNT^[i].fi3_pathname);
SubItems.Add(FileInfoNT^[i].fi3_username);
end;
end;

```

Тут Caption є ідентифікатором, поле fi3_pathname означає шлях до файлу, а поле fi3_username – ім'я користувача. Закриття файлу відбувається за допомогою TMainForm.btnCloseFC.

```

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));

```

Вхідний та вихідний трафік визначаються за допомогою процедури TMainForm.tmrTT. Спершу вводиться допоміжна функція, що перетворює MAC адресу до традиційного виду.

```

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;

```

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

```
end;
```

Визначаємо спеціальний тип, щоб можна було передати у функцію масив `type TMAC = array [0..7] of Byte`. Як перше значення приймається масив, друге значення – розмір даних у масиві. В самій процедурі `TMainForm.tmrTT` використовуються наступні змінні.

```
FLibHandle : THandle;  
Table: TMibIfTable;  
i : integer;  
Size : integer;
```

Тут таблиця `Table` має тип `TMibIfTable`, який описується таким чином.

```
PTMibIfRow = ^TMibIfRow;  
TMibIfRow = packed record  
    wszName: array[1..MAX_INTERFACE_NAME_LEN] of WCHAR;  
    dwIndex: DWORD;  
    dwType: DWORD;  
    dwMTU: DWORD;  
    dwSpeed: DWORD;  
    dwPhysAddrLen: DWORD;  
    bPhysAddr: array[1..MAXLEN_PHYSADDR] of byte;  
    dwAdminStatus: DWORD;  
    dwOperStatus: DWORD;    dwLastChange: DWORD;  
    dwInOctets: 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;
```

При визначенні вхідного-вихідного трафіку після вимкнення таймера виконуються наступні операції по визначенню кількості відправлених та прийнятих байт.

```
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
```

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57


```

ShareNT.shi2_type := STYPE_DISKTREE;
ShareNT.shi2_remark := ' ' ;
ShareNT.shi2_permissions := ACCESS_ALL;
ShareNT.shi2_max_uses := DWORD(-1);
ShareNT.shi2_current_uses := 0;
GetMem(TmpDirNT, TmpLength);
StringToWideChar(TmpDir, TmpDirNT, TmpLength);
ShareNT.shi2_path := TmpDirNT;
ShareNT.shi2_passwd := ' ' ;

```

Після цих операцій додаємо ресурс та звільняємо пам'ять.

```

NetShareAddNT(nil, 2, @ShareNT, nil);
FreeMem (TmpName7);
FreeMem (TmpDirNT);

```

Аналогічні операції по додаванню ресурсу проводяться, якщо операційна система відноситься до сімейства Windows 10/11. В цьому випадку основною змінною є Share9x типу TShareInfo50.

```

FLibHandle := LoadLibrary(' SVRAPI.DLL' );
if FLibHandle = 0 then Exit;
@NetShareAdd := GetProcAddress(FLibHandle, ' NetShareAdd' );
if not Assigned(NetShareAdd) then
begin FreeLibrary(FLibHandle); Exit; 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_FULLL; //Доступ
FillChar(Share9x.shi50_remark,
SizeOf(Share9x.shi50_remark), #0); //Коментар
FillChar(Share9x.shi50_path,
SizeOf(Share9x.shi50_path), #0);
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);

```

					БКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм Lucifer. Алгоритм Lucifer являє собою мережу перестановок і підстановок, його основні блоки нагадують блоки алгоритму DES. В DES результат функції f складається операцією XOR із входом попереднього раунду, утворюючи вхід наступного раунду. В S-блоках алгоритму Lucifer 4-бітові входи й виходи, вхід S-блоків являє собою перетасований вихід S-блоків попереднього раунду, входом S-блоків першого раунду служить відкритий текст. Для вибору використовуваного S-блоку із двох можливих використовується біт ключа. (Lucifer реалізує все це в єдиному T-блоці з 9 бітами на вході й 8 бітами на виході). На відміну від алгоритму DES, половини блоку між раундами не переставляються, та й саме поняття половини блоку в алгоритмі Lucifer не використовується. У цього алгоритму 16 раундів, 128-бітові блоки й більше проста, чим в DES, схема розгорнення ключа.

Блок тексту розглядається як ненегативне ціле число, або як кілька незалежних ненегативних цілих чисел. Довжина блоку завжди вибирається рівною ступеню двійки. У алгоритмі Lucifer використовуються наступні типи операцій:

- Таблична підстановка, при якій група біт відображається в іншу групу біт. Це так звані S-бок.
- Переміщення, за допомогою якого біти повідомлення переупорядковуються.
- Операція додавання по модулю 2, позначувана XOR або \oplus .
- Операція додавання по модулю 2^{32} або по модулю 2^{16} .
- Циклічне зрушення на деяке число біт.

Ці операції циклічно повторюються в алгоритмі, створюючи так звані раунди. Входом кожного раунду є вихід попереднього раунду й ключ, що отриманий по певному алгоритму із ключа шифрування K . Ключ раунду

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

називається підключем. Алгоритм шифрування може бути представлений у такий спосіб:

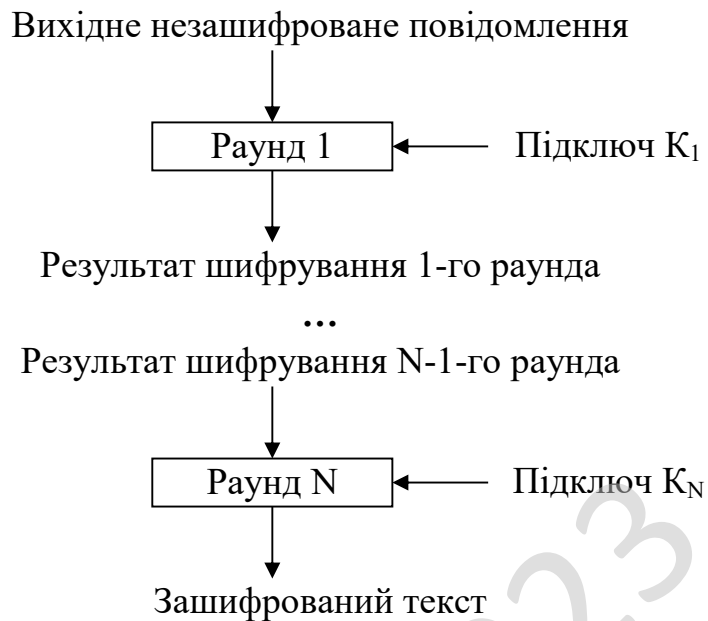


Рисунок 4.3 – Структура алгоритму алгоритмі Lucifer

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

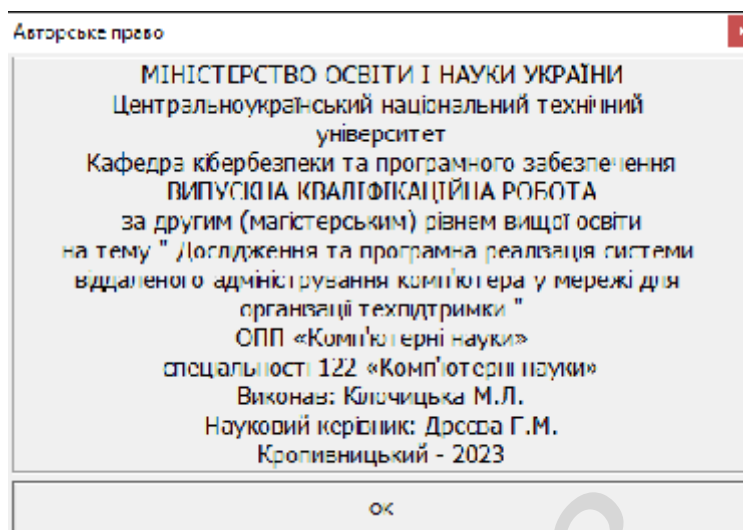


Рисунок 5.2 – Авторське право

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 «безплатне програмне забезпечення» та «вільне програмне забезпечення», хоча вони суттєво відрізняються.

Безплатне програмне забезпечення можна безоплатно встановлювати та використовувати (іноді з певними обмеженнями, як, наприклад, «безплатне 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-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки.

Метою розробки є дослідження та програмна реалізація системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки.

Об'єктом дослідження є процес віддаленого адміністрування комп'ютера у мережі для організації техпідтримки.

Предметом дослідження є методи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки.

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

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

– Удосконалено метод віддаленого адміністрування комп'ютера у мережі для організації техпідтримки.

– Розроблено вітчизняний продукт віддаленого адміністрування комп'ютера у мережі для організації техпідтримки, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

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 віддаленого адміністрування комп'ютера у мережі для організації техпідтримки. 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	260
3. Запланований термін розробки, днів	Frq	48 (2 місяць)
4. Група задачі підсистеми управління (1-6)	–	1
5. Ступінь новизни задачі (А, Б, В, Г)	–	В
6. Складність алгоритму (1, 2, 3)	–	2

Продовження таблиці 7.1

1	2	3
7. Кількість макетів вхідної інформації	–	8
8. Кількість форм вихідної інформації.	–	6
9. Мова програмування (1-6)	–	2
10. Попередній досвід (1-6)	–	3
11. Гнучкість проекту ПП (1-6)	–	3
12. Детальність проекту ПП (1-6)	–	1
13. Рівень спрацьованості колективу (1-6)	–	2
14. Ступінь вимірності процесів (1-6)	–	3
15. Необхідна надійність програмного забезпечення (1-6)	–	3
16. Розмір бази даних (порівняно з розміром програми) (1-6)	–	4
17. Складність кінцевого програмного продукту (1-6)	–	5
18. Необхідний рівень забезпечення повторного використання (1-6)	–	2
19. Документованість відповідно до планованого життєвого циклу (1-6)	–	3
20. Вимоги до швидкодії ПП (1-6)	–	3
21. Обмеження на розміри основного сховища даних (1-6)	–	2
22. Різноманітність використовуваних обчислювальних платформ (1-6)	–	4
23. Професійний рівень аналітиків (1-6)	–	3
24. Професійний рівень програмістів (1-6)	–	4
25. Постійність складу команди розробників (1-6)	–	2
26. Досвід розробки додатків (1-6)	–	1
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)	–	2
29. Досвід роботи з програмними інструментами розробки (1-6)	–	3
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.	–	260000
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	40
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-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

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.

$$B = 1,01 + 0,001(2,43 + 3,64 + 4,22 + 3,95 + 2,73) = 1,027.$$

$$T_{ном} = 2,45 \cdot 2,2^{1,027} = 5,5 \text{ люд-міс.}$$

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.

$$T_{уточн} = 5,5 \cdot (1 \cdot 1,09 \cdot 1,30 \cdot 0,91 \cdot 1 \cdot 1 \cdot 1 \cdot 1,15 \cdot 1 \cdot 0,87 \cdot 1,10 \cdot 1,22 \cdot 1,12 \cdot 1,10 \cdot 1 \cdot 1 \cdot 1,10) = 12,9 \text{ люд-міс.}$$

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);

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

$$T_{РП} = 0,3 \cdot 2,66 \cdot 12,9^{0,33+0,2(1,027-1,01)} \cdot 100 = 131 \text{ люд/день.}$$

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

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

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	131	Ф 7.1-7.4
Впровадження	15	Д13
Всього	180	–

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.

$$Ч = \frac{180 \cdot 1}{24 - 3} = 8,6 \text{ ставки.}$$

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	70	175	3
Кабельне господарство електромережі	48	50	2400	40
Копіювальний апарат	285	1	285	5
Усього за рік:			3 _ч	221

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:

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

$$\Phi_{op}^c = \frac{221 \cdot 1}{1,2} = 184,1 \text{ год.}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

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

$$Ч_{ел} = 184,1 / (24 \cdot 8) = 1 \text{ ставки.}$$

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	2	0,5
	Підтримка постійних клієнтів	1	
	Оформлення договорів, ведення тендерів	0,5	
	Контроль взаєморозрахунків з постачальниками	0,5	
Всього		4	
Дизайнер WEB	Розробка концепції оформлення та інтерфейсу сайту, оптимізація дизайну існуючих, проектує їх структуру та навігацію	2	0,5
	Створення графічних і стилістичних елементів сайту	1	
	Оформлення банерів і промо-сторінок	0,5	
	Розміщення графіки і контенту на Інтернет сторінках	0,5	
Всього		4	
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	17000	34000
Продакт-менеджер	0,5	15000	15000
Інженер-програміст	8,6	16000	275200
Інженер-електронщик	1	15000	30000
Інженер-системотехнік	0,25	15000	7500
Адміністратор мережі	0,5	15000	15000
Системний програміст	0,25	15000	7500
Дизайнер WEB	0,5	16000	16000
Інженер-верстальник	0,25	15000	7500
Бухгалтер-економіст	0,5	16000	16000
Всього за період розробки	$R_{cn} = 13,35$	-	$\Phi_{роб} = 423700$

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.

$$z_{cd} = \frac{423700}{13,35 \cdot 48} = 661 \text{ грн.}$$

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-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

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

де: R_{cn}^1 – кількість робочих місць виконавців, шт. Приймаємо 12 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:

$$B_{y\partial} = 12 \cdot 8 \cdot 20000 = 1920000 \text{ грн.}$$

Вартість передавальних пристроїв складає 10% від вартості будівель, і у даному випадку вона складе: 192000 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.

$$I_{не} = 12 \cdot 3500 = 42000 \text{ грн.}$$

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 Комп'ютерторг за 25.10.23 – джерело <http://computorg.ua/price.html>

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

Продовження таблиці 7.6

Найменування комплектуючої або обладнання	Тип	Оптова ціна
інше	Клавіатура, мишка	Подарунок
Монітор	22" TFT, ASUS VW223D (5ms, 300/3000:1, 170/160, D-SUB, Wide)	3200
Принтер лазерний	Canon i-SENSYS LBP6030W	2700
Принтер струминний	Epson Stylus Photo P50 (C11CA45341) + USB cable	5500
Сканер	Epson Perfection V37 Photo	2970
Копіювальний апарат	Canon i-SENSYS MF217W with Wi-Fi	5965
Пристрій безперебійного живлення	UPS APC BACK-UPS ES 525VA 230V RUSSIA (BE525-RS)	1496

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

Для визначення необхідної кількості капітальних вкладень складемо 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	12	11186	13423,2	147655,2
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050

Продовження таблиці 7.7

Найменування обчислювальної техніки	Кількість, шт.	Ціна за одиницю, грн.	Витрати на транспортування, монтаж та випробовування.	Загальна вартість, грн.
Сканери	1	2970	297	3267
Копіюв. апарат	1	5965	596,5	6561,5
Всього	–	–	–	169473,7

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	1920000	-	-
2. Передавальні пристрої	192000	-	-
Всього по групі	2112000	5	105600
Група 4			
3. Обчислювальна техніка	169474	-	-
Всього по групі	169474	50	84737
Нематеріальні активи			
4. Нематеріальні активи	260000	10	26000

Продовження таблиці 7.8

1	2	3	4
Група 5, 6			
5. Вимірювальні пристрої	5190	25	1297,5
6. Транспортні засоби	72500	20	14500
7. Господарський інвентар	42000	25	10500
Всього по групі	119690	-	26297,5
Разом	$K_p = 2661164$		$A_p = 242634,5$

Примітка: вартість автомобіля взята по даним з автосалону автотрейдинг, вкладки автобазар, джерело <http://www.auto-trading.com.ua/sale/lot20772.html>, складає 1960 USD, що враховуючи прийнятий для розрахунку курс 37 складає 72500 грн.

7.5 Визначення собівартості розробки та ціни програмної продукції

Визначимо основну зарплату виконавців:

$$Z_o = \frac{Z_{cd} \cdot T_{nz}}{N_e}, \quad (7.11)$$

де: N_e – кількість екземплярів програм, шт.

$$Z_o = 661 \cdot 180 / 260 = 458 \text{ грн.}$$

Визначимо додаткову зарплату (оплата відпусток, виконання державних та суспільних обов'язків) на рівні 10%:

$$Z_d = Z_o \cdot H_q \cdot 0,01, \quad (7.12)$$

де: H_q – норматив додаткової зарплати, %.

$$Z_d = 458 \cdot 10 \cdot 0,01 = 46 \text{ грн.}$$

Відрахування на соціальні потреби за нормативом $H_c = 22\%$ від суми основної та додаткової зарплати:

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

$$C_{oc} = 0,01 \cdot H_c (Z_o + Z_d), \quad (7.13)$$

де: H_c – відрахування на соціальні потреби, %.

$$C_{oc} = 0,01 \cdot 22(458+46) = 111 \text{ грн.}$$

Визначимо загальногосподарські витрати (електроенергію, ремонт і утримання приміщень і т.д) за нормативом $H_z = 15\%$ від основної зарплати:

$$G_{ocn} = Z_o \cdot H_z \cdot 0,01, \quad (7.14)$$

де: H_z – загальногосподарські витрати, %.

$$G_{ocn} = 458 \cdot 15 \cdot 0,01 = 69 \text{ грн.}$$

Визначимо витрати на матеріали для розробки програмної продукції за нормами споживання та діючими цінами за одиницю виміру:

$$Z_M = (Z_{M1} + Z_{M2} + Z_{M3})/N_e, \quad (7.15)$$

де: Z_{M1} – вартість паперу, грн.; Z_{M2} – вартість запам'ятовуючих пристроїв, грн.; Z_{M3} – вартість фарби, картриджів, тонеру, грн.; N_e – кількість екземплярів програм, шт.

Згідно виданих викладачем норм приймаємо одну пачку паперу на місяць розробки. Тоді, враховуючи, що вартість пачки паперу складає $U_n = 120$ грн., визначаємо вартість паперу за період розробки $N_m = 2$ міс:

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

$$Z_{M1} = 120 \cdot 2 = 240 \text{ грн.}$$

Згідно виданих викладачем норм до вартості запам'ятовуючих пристроїв входить вартість CD дисків в кількості, що дорівнює кількості екземплярів програм та одного DVD диска для збереження резервної копії програми:

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

де: U_d – вартість дисків CD/DVD: CDR TDK 700Mb, 80Min, 52x Cake box – 13 грн./шт., DVD-R LG 4,7Gb, 16x speed Cake box – 13 грн./шт.

$$Z_{M2} = 260 \cdot 13 + 13 = 3393 \text{ грн.}$$

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

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

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

де: C_3 – 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.

$$Z_{M3} = 574 + 558 + 570 = 1702 \text{ грн.}$$

$$Z_M = (240 + 3393 + 1702) / 260 = 21 \text{ грн.}$$

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, %.

$$O_n = 458 \cdot 15 \cdot 0,01 = 69 \text{ грн.}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_e = 260$ прим.):

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

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

$$A_m = 242634 \cdot 2 / (260 \cdot 12) = 156 \text{ грн.}$$

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

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

$$C_n = 458 + 46 + 111 + 69 + 21 + 69 + 156 = 930 \text{ грн.}$$

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 40%.

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

де: P_n – рівень рентабельності, %.

$$P_p = 0,01 \cdot 40 \cdot 930 = 379 \text{ грн.}$$

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		80

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	458
2. Додаткова зарплата виконавців	Z_d	46
3. Відрахування на соціальні потреби	C_{oc}	111
4. Загальногосподарські витрати	Γ_{ocn}	69
5. Витрати на матеріали	Z_M	21
6. Освоєння нових операційних систем, мов програмування	O_n	69
7. Амортизація основних фондів	A_M	156
8. Повна собівартість програмного забезпечення	C_n	930
9. Плановий прибуток	P_p	379
10. Ціна підприємства $C_n = C_n + P_p$	C_n	1309
11. Податок на додану вартість $ПДВ = 0.01 \cdot H_{ов} \cdot C_n$	$ПДВ$	261,8
12. Відпускна ціна програмної продукції $C = C_n + ПДВ$	C	1570,8

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

Після купівлі нового програмного забезпечення кількість профілактичних годин робіт зменшилася з 250 годин на рік до 140 годин на рік, тому витрати на технічне обслуговування зменшилися з:

$$Z_{p\text{ баз}} = 250 \cdot 130 \cdot 1,1 \cdot 1,22 = 43615 \text{ грн},$$

до:

$$Z_{p\text{ нов}} = 140 \cdot 130 \cdot 1,1 \cdot 1,22 = 24424 \text{ грн}.$$

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 ($C_{ел}$):

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

$Z_{ел\text{ баз}} = Z_{ел\text{ нов}}$. Витрати на електроенергію не змінюються.

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	50	–	1571	–	785,5
Всього відрахувань	-	–	1571	–	785,5

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:

$$E_e = (C_n - C_n) \cdot N_e - \sum_{i=1}^m E_{P_m} \cdot K_{P_m}, \quad (7.25)$$

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

$$E_v = (1309 - 930) \cdot 260 - (0,05 \cdot 2112000 + 0,5 \cdot 169474 + 0,25 \cdot 47190 + 0,2 \cdot 72500 + 0,1 \cdot 260000) \cdot 2/12 = 58101 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у виробника програмної продукції:

$$T_e = \frac{K_p^*}{(C_n - C_n) \cdot N_e}, \quad (7.26)$$

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.

$$T_e = \frac{549164}{(1309 - 930) \cdot 260 \cdot 12 / 2} = 0,93 \text{ року.}$$

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.

$$E_{cn} = (43615 - 25210) - 0,5 \cdot 1571 = 17620 \text{ грн.}$$

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

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

$$T_{cn} = \frac{1571}{43615 - 25210} < 0,1 \text{ року.}$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1. Кількість екземплярів програми	Прим.	260
2. Повна собівартість розробленої програми	Грн.	930
3. Ціна розробленої програми	Грн.	1309
4. Плановий прибуток від реалізації розробленої програми	Грн.	379
5. Рентабельність програмної продукції	%	40
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	2661164
7. Загальний прибуток від реалізації програмної продукції	Грн.	98540
8. Величина економічного ефекту при виготовленні програмної продукції	Грн.	58101
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років	0,93
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	1571
11. Величина економічного ефекту у користувача програмної продукції	Грн.	17620
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	0,1

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.

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

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

Ці шкідливі фактори можуть привести до професійних захворювань.

До недоліків умов праці користувачів комп'ютерної техніки можна віднести:

- недостатню площу і обсяг виробничого приміщення;
- недотримання вимог, мікроклімату на робочих місцях;
- низький рівень освітленості у приміщеннях і на робочих поверхнях апаратури;
- підвищений рівень низькочастотних магнітних полів від моніторів;
- порушення вимог організації робочих місць;
- недотримання вимог до режимам праці та відпочинку;
- надмірне виробничу навантаження працівників;
- відсутність навичок зниження впливу психоемоційного напруги.

Відповідно до ст.14 Закону «Про охорони праці» [3] на роботодавця покладено обов'язок забезпечити: безпеку працівників при експлуатації устаткування; застосування коштів індивідуальної захисту працівників; відповідні вимоги охорони праці, умови праці в кожному робоче місце; дотримання режиму праці та відпочинку працівників; навчання безпечним методам і прийомам

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		86

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 [5].

Шкідливі і небезпечні фактори при роботі з комп'ютером

Електронно-обчислювальні машини (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 [2].

Шкідливими факторами при роботі з персональним комп'ютером є неонізуюче випромінювання промислової частоти, збільшене нервово-емоційне навантаження на оператора, збільшення навантаження на органи зору та дрібні стереостатичні рухи кінцівок.

Ці фактори можуть викликати у працівника певні розлади здоров'я, зокрема підвищення артеріального тиску, кон'юктивіти, тендовагініти ті інші захворювання.

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

Через наявність зазначених факторів працівники, які працюють з персональними комп'ютерами, підлягають попередньому та періодичному медичному огляду згідно з пунктом 6.2.3 додатку 4 до наказу Міністерства охорони здоров'я України "Про затвердження Порядку проведення медичних оглядів працівників певних категорій" від 21 травня 2007 року №246 [8].

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		87

8.2 Аналіз умов праці

Приміщення розташовано на третьому поверсі п'ятиповерхового будинку. У приміщенні розташовано 3 робочих місць з комп'ютерами (далі ПК). Відповідно до норм «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98 [4] площа, що відводиться для робочого місця з комп'ютером повинна бути не менше 6 м², об'єм не менше 20 м³. Розміри даного приміщень складають: довжина – 6 м, ширина – 4,5 м, висота – 3,5 м, тобто загальна фактична площа складає 27 м². Необхідна площа на 3 робочих місця із установленими ПК складає 18 м², що не перевищує фактичну. Обсяг кабінету на одного працюючого складає 31,5м³, отже відповідає нормі ДСанПіН 3.3.2-007-98 – не менше 20 м³.

При роботі з ПК людина може піддатися впливу шкідливих та небезпечних факторів. Під шкідливими виробничими факторами розуміють фактори, тривалий вплив яких викликає розвиток професійних захворювань. Небезпечні виробничі фактори – вплив яких на працюючого викликає травму, тобто пошкодження організму. Шкідливі і небезпечні чинники, з якими стикається бібліограф при роботі з ПК, приведені в таблиці 8.1.

По категорії вибухо – і пожежонебезпеки, згідно дане приміщення відноситься до категорії В – пожежонебезпечне, тому що присутні тверді матеріали, що горять, такі як дерев'яні столи, папір і інше. Виходячи з категорії пожежонебезпеки і поверховості будинку, ступінь вогнестійкості будівлі II. Згідно з ДБН В 1.1-7-2016 «Пожежна безпека об'єктів будівництва» [5] ЕОМ повинні розташовуватись в будівлі не менше ніж II ступню вогнестійкості.

По ступені небезпеки поразки людей електричним струмом відділ, згідно, класифікується як приміщення з підвищеною небезпекою, тому що не виключена можливість одночасного дотику людини до маючих з'єднання з землею конструкціям будинку, з одного боку, і до металевих корпусів електроустаткування, що можуть виявити під напругою – з іншого.

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		88

Таблиця 8.1 – Перелік шкідливих та небезпечних виробничих факторів

Найменування факторів	Можливі джерела їх виникнення	Характер дії
Небезпека ураження електричним струмом	Мережа живлення	Небезпечний
Пожежонебезпечність приміщень	Наявність матеріалів, що згорають і джерел запалення (електроапаратура)	Небезпечний та шкідливий
Іонізація повітря	Статична електрика випромінювання	Шкідливий
Підвищений рівень шуму	Шум створюється перетворювачем напруги ЕОМ, її технічною периферією, а також людьми, що працюють в приміщенні	Шкідливий
Несприятлива освітленість	Недостатнє штучне і природне освітлення	Шкідливий
Незадовільні параметри мікроклімату	Незадовільний стан системи опалення і вентиляції	Шкідливий
Психофізіологічні напруження	Монотонність праці, перенапруженість зорових аналізаторів, розумова напруженість, незручність і статичність пози	Шкідливий

Для забезпечення вищевказаних оптимальних метеорологічних умов у помешканні передбачена система опалення (загальне парове) в холодному періоді, та вентиляція і кондиціонування в теплий період року, згідно ДБН2.5–67–2013 «Опалення, вентиляція та кондиціонування» [6]. При виконанні замірів параметрів мікроклімату, значення їх відповідали оптимальним та допустимим параметрам

відповідно до ДСанПіНЗ.3.2.007 – 98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно – обчислювальних машин».

Припустимий рівень іонізації повітря помешкання відповідно до СН 21.52-80 повинен складати 1500 – 3000 один./м3.

Нормування освітлення здійснюється відповідно до ДБН В.2.5 – 28 – 2006 «Природне та штучне освітлення». [7]

Відділ забезпечений комбінованим освітленням. В темний час доби передбачається загальне і/або місцеве рівномірне штучне, а в світлий – бокове одностороннє природне освітлення два віконних прорізи.

Одним з найбільш поширеніших чинників зовнішнього середовища, який несприятливо впливає на людину, є шум. Вплив шуму на організм людини залежить від рівня звукового тиску, частотних характеристик, тривалості дії, а також індивідуальних особливостей людини.

При тривалій дії шуму у оператора ЕОМ виявляються симптоми утомленості, нервового збудження, що сприяють погіршенню працездатності і допущенні помилок при роботі. Для уникнення шкідливої дії шуму на організм працюючого, необхідне дотримання нормованих параметрів, які не повинні перевищувати допустимих величин. При роботі на комп'ютері рівень шуму не повинен перевищувати 50 дБА. Приміщення розташоване вікнами у двір і знаходиться далеко від проїжджої частини вулиці. Основними джерелами шуму в приміщенні є устаткування і люди. Розглянута кімната не призначена для прийому відвідувачів і тому в ній не спостерігається великого скупчення людей. Тому основним джерелом шуму є комп'ютерна техніка.

Джерелами шуму при роботі ЕОМ є механічні частини принтера, що рухаються, і вентилятори (Lпк = 35 дБА, Lрпн = 48 дБА.) При роботі вентиляційної системи, що забезпечує оптимальний температурний режим електронних блоків ЕОМ і вмонтована в задню панель, створюється аеродинамічний шум. Шум, створюваний працюючим комп'ютером, може бути охарактеризований як широко

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

смуговий постійний з аперіодичним посиленням при роботі принтера. Час роботи ПЕОМ – 6 – 8 год. за добу; принтери працюють не більш 1,5-2 год. за добу.

При наявності великої кількості джерел шуму еквівалентне значення шуму $L_{ЭКВ}$, дБА розраховують по наступній формулі:

$$L_{ЭКВ} = 10 \cdot \lg \left(\frac{1}{T} \sum_{i=1}^n \left(t_i \cdot 10^{0.1 L_i} \right) \right) \quad (8.1),$$

де L_i – рівень шуму i -го джерела (пристрою),

t_i – час роботи i -го джерела (пристрою),

T – загальний час роботи,

n – кількість джерел шуму даного типу;

Для даного приміщення необхідні змінні складають:

Загальний час роботи – робітник день, тобто $T=8$ годин.

Для фонового шуму (вентиляторів):

$$L_1 = 35 \text{ дБА}, T_1 = 8 \text{ годин}, n_1=15 (5 \times 3);$$

Для лазерного принтера Lexmark Jet:

$$L_2 = 48 \text{ дБА}, T_2 = 2 \text{ години}, n_2=1, \text{ для сканера } L_3 = 46 \text{ дБА}, T_3 = 2 \text{ години}.$$

Підставляємо отримані величини у формулу (8.1):

$$L_{ЭКВ} = 10 \cdot \lg \left(\frac{1}{8} \cdot (15 \cdot 8 \cdot 10^{6.135} + 1 \cdot 2 \cdot 10^{6.148} + 1 \cdot 2 \cdot 10^{6.148}) \right) = 46,3 \text{ дБА}$$

Таким чином, еквівалентний рівень шуму в приміщенні за робітник день $L_{ЭКВ} = 46,3$ дБА, тобто не перевищує норму 50 дБА.

8.3 Техніка безпеки та протипожежна профілактика

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 проводки

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

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 1м від підлоги. Усього до складу входять п'ять розеток і дві клеми заземлення. Всі обчислювальні машини з'єднані із клемами 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 пожежобезпечність 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” [8], 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. Відповідно правилам пристрою електроустановок ЕОМ можна віднести до електроустановок з робочою напругою до 1000 В.

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

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.

8.4 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 $d=0.06 \text{ м}$ і довжиною $l = 1.7 \text{ м}$;
- конструкція заземлювача – розташування заземлювачів по контуру. Розрахунок проводимо за стандартною методикою [7].

Визначимо розрахунковий опір землі:

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

Визначаємо потрібну кількість заземлювачів:

$$n' = \frac{R_g}{R_{zn}} = \frac{45.17}{10} = 4.5 \approx 5 \text{ шт.}$$

Коефіцієнт використання вертикальних заземлювачів враховує ефект екранування. При вибраному значенні $k = a/l$, де a – відстань між вертикальними заземлювачами, м; $k = 1$ при $a = 2.4$ м. Таким чином коефіцієнт використання вертикального заземлювача за довідковими даними дорівнює $\eta_B = 0.6$.

Кількість вертикальних заземлювачів з урахуванням коефіцієнту використання η_B приблизно складає

$$n = \frac{R_g}{R_{zn} \cdot \eta_B} = \frac{45.17}{10 \cdot 0.6} = 7.53 \approx 8 \text{ шт.}$$

Довжина горизонтального заземлювача, необхідна для розміщення вертикальних заземлювачів по контуру

$$L = a \cdot n = 2.4 \cdot 8 = 19.2 \text{ м}$$

Опір горизонтального заземлювача R_G , Ом, прокладеного на глибині $h = 0.65$ м від поверхні землі буде

$$R_G = \frac{R_{pz}}{2 \cdot 3.14 \cdot L} \cdot \ln \frac{2 \cdot L^2}{b \cdot h} = \frac{110}{2 \cdot 3.14 \cdot 19.2} \cdot \ln \frac{2 \cdot 19.2^2}{0.06 \cdot 0.65} = 10.61 \text{ Ом}$$

де $b = 0.04$ м – ширина сталевий смуги, з якої виготовлений заземлювач.

Обчислюємо загальний опір:

$$R_z = \frac{R_B \cdot R_G}{n \cdot R_G \cdot \eta_B + R_g \cdot \eta_g} = \frac{45.17 \cdot 10.61}{6 \cdot 10.61 \cdot 0.6 + 45.17 \cdot 0.34} = 8.33 \text{ Ом}$$

де η_g – коефіцієнт використання горизонтального заземлювача ($\eta_g = 0.34$).

Маємо $8.33 \text{ Ом} < 10 \text{ Ом}$, отже нормативне обмеження $R_z < R_{z, \text{норм}}$ виконується.

8.5 Список використаних джерел інформації

1. Державні будівельні норми України ДБН В.2.5 – 28 – 2006: Інженерне обладнання будинків і споруд. Природне і штучне освітлення. – Режим доступу до

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		95

ресурсу: <https://zakon.rada.gov.ua/rada/show/v0168667-06#Text> (дата звернення 19.10.22).

2. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин: ДСанПІН 3.3.2-007-98. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/v0007282-98> (дата звернення 19.10.22).

3. Закон України «Про охорону праці» від 14.10.1992 р. № 2694-ХІІ. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/2694-12> (дата звернення 19.10.22).

4. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин: ДСанПІН 3.3.2-007-98. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/v0007282-98> (дата звернення 19.10.22).

5. Наказ Міністерства регіонального розвитку, будівництва та житлово-комунального господарства України 31.10.2016 «Про затвердження ДБН В.1.1-702016» – Режим доступу до ресурсу: <https://ips.ligazakon.net/document/fn025551>

6. Наказ Міністерства регіонального розвитку, будівництва та житлово-комунального господарства України 25.01.2013 №24 «Про затвердження ДБН В.2.5-67:2013 "Опалення, вентиляція та кондиціонування" – Режим доступу до ресурсу <https://zakon.rada.gov.ua/rada/show/v0024858-13#Text>

7. Державні будівельні норми України: ДБН В.2.5-28:2018. – Режим доступу до ресурсу: <https://goo.su/9AkQ> (дата звернення 19.09.22).

8. 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 19.09.22).

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,

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		96

Центральноукраїн. 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).

10. Охорона праці. Ч. 1. Захисне заземлення: метод. вказ. до викон.
розрахунків з викор. персон. ЕОМ IBM сумісного типу / Кіровоград. ін-т с.-г.
машинобуд.; [укл. О. В. Оришака, Є. К. Солових, В. О. Оришака]. – Кіровоград :
КІСМ, 1997. – 20 с. – Режим доступу:
<http://dspace.kntu.kr.ua/jspui/handle/123456789/4358> (дата звернення 19.09.23).

КБПЗ – 2023

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки.

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 віддаленого адміністрування комп'ютера у мережі для організації техпідтримки.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем віддаленого адміністрування комп'ютера у мережі для організації техпідтримки.
- Досліджена система віддаленого адміністрування комп'ютера у мережі для організації техпідтримки.
- На основі отриманих результатів досліджень створена програмна реалізація системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання віддаленого адміністрування комп'ютера у мережі для організації техпідтримки.

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-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		98

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 RAD Studio Delphi. 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 Lucifer.

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 17620 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 0,1 роки.

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		99

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кілочичька М.Л. Дослідження та програмна реалізація системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2023.
2. Russ White & Ethan Banks «Computer Networking Problems and Solutions: An Innovative Approach to Building Resilient, Modern Networks». 2017. – 832 p.
3. 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.
4. 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.
5. 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.
6. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». *CEUR Workshop Proceedings*. Volume 2740, 2020, Pages 102-114.
7. 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.

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		100

8. 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.

9. 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.

10. 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.

11. 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.

12. 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.

13. 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.

14. 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.

15. 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.

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

17. 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.

18. 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.

19. 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.

20. 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.

21. 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.

22. 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.

23. 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.

24. 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.

25. 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.

26. 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.

27. 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.

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

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		103

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

29. 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.

30. 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.

31. 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.

32. 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.

33. 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.

34. 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.

35. 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).

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

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		104

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

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

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

39. 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.

40. 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.

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

42. 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.

43. 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.

					ВКРМ-122.23.0067.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		105

44. 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.

45. 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.

46. 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.

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

50. 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 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.

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-122.23.0067.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Кілочицька М.Л.				<i>Дослідження та програмна реалізація системи віддаленого адміністрування комп'ютера у мережі для організації техпідтримки</i>	Літ.	Аркуш	Аркушів
Перевірів	Дресва Г.М.					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 віддаленого адміністрування комп'ютера у мережі для організації техпідтримки.

2 Підстава для розробки

Підставою для розробки служить завдання на випускну 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 віддаленого адміністрування комп'ютера у мережі для організації техпідтримки.

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-122.23.0067.00.00.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 віддаленого адміністрування комп'ютера у мережі для організації техпідтримки;
- 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-122.23.0067.00.00.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 RAD Studio Delphi.

					ВКРМ-122.23.0067.00.00.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 повинна бути розглянута техніка безпеки та протипожежна профілактика.

					ВКРМ-122.23.0067.00.00.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 – 106 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 15.12.2023 р.

					ВКРМ-122.23.0067.00.00.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

_____ Дреєва Г.М.

*Дослідження та програмна реалізація
системи віддаленого адміністрування комп'ютера у мережі для організації
техпідтримки*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 49

Літера: РП

Кропивницький – 2023 року

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 Клієнт }
    ( 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
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 останій 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 ; // дані
                                    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;
        //
        with ICMPStats.OutStats do
        begin
            ICMPOut.Add( ' Повідомлення вправлено      : ' + 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
    else
        IcmpIn.Add( SysErrorMessage( ErrorCode ) );
    Dispose( ICMPStats );
end;

```

```
end;

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

initialization

  RecentIPs := TStringList.Create;

finalization

  RecentIPs.Free;

end.
```

K6П3_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.
```

Основна програма

Файл Monitor_Local_Net. dpr основної програми

```
program Monitor_Local_Net;

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);
```

```

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;
    dwInNUCcastPkts     : DWORD;
    dwInDiscards         : DWORD;
    dwInErrors           : DWORD;
    dwInUnknownProtos   : DWORD;
    dwOutOctets          : DWORD;
    dwOutUcastPkts      : DWORD;
    dwOutNUCcastPkts    : 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
    //

```

```

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 списку 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;
  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.
```

К6ПЗ_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;
    dwInNUCastPkts: 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: PTIP_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-CTPYKTYPA -----

PTMibICMPStats = ^TMibICMPStats;
TMibICMPStats = packed record
    dwMsgs: DWORD;
    dwErrors: DWORD;
    dwDestUnreachs: DWORD;
    dwTimeEcxcds: DWORD;
    dwParmProbs: DWORD;
    dwSrcQuenchs: 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.
```

K6П3_2023

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.
```