

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

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

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи менеджменту
файлів ОС Windows 11/Windows Server 2022”

КБПЗ - 2023

Виконав здобувач вищої освіти
II курсу, групи КН-22М-1
ОПП «Комп’ютерні науки»
спеціальності 122 «Комп’ютерні науки»
_____ Дудченко А.В.
« ____ » _____ 2023 р.

Керівник проекту
кандидат технічних наук, доцент
_____ Доренський О.П.
« ____ » _____ 2023 р.
Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
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

Дудченко Анастасії Василівні

(прізвище, ім'я, по батькові)

- | | | | | | | | | | | | |
|--|--|--|----------------------------|---|---|--|--|--|---------------------|--|--|
| 1. Тема роботи | <u>Дослідження та програмна реалізація системи менеджменту файлів ОС Windows 11/Windows Server 2022</u> | | | | | | | | | | |
| 2. Керівник роботи | <u>Доренський Олександр Павлович, канд. техн. наук, доцент</u>
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу № 32-13 від 04.08.2023 року | | | | | | | | | | |
| 3. Строк подання студентом роботи до захисту | <u>10.12.2023 р.</u> | | | | | | | | | | |
| 4. Мета та завдання випускної кваліфікаційної роботи: | <u>Метою розробки є дослідження та програмна реалізація системи менеджменту файлів ОС Windows 11/Windows Server 2022</u> | | | | | | | | | | |
| 5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) | <table border="1"><tr><td><u>1. Призначення та область використання.</u></td><td><u>6. Наукова новизна.</u></td></tr><tr><td><u>2. Перегляд аналогічних існуючих систем.</u></td><td><u>7. Економічна ефективність розробленої програми.</u></td></tr><tr><td><u>3. Опис і обґрунтування проектних рішень.</u></td><td><u>8. Заходи з охорони праці та техніки безпеки.</u></td></tr><tr><td><u>4. Етапи програмування системи.</u></td><td><u>9. Висновки.</u></td></tr><tr><td><u>5. Впровадження системи в промислову експлуатацію</u></td><td></td></tr></table> | <u>1. Призначення та область використання.</u> | <u>6. Наукова новизна.</u> | <u>2. Перегляд аналогічних існуючих систем.</u> | <u>7. Економічна ефективність розробленої програми.</u> | <u>3. Опис і обґрунтування проектних рішень.</u> | <u>8. Заходи з охорони праці та техніки безпеки.</u> | <u>4. Етапи програмування системи.</u> | <u>9. Висновки.</u> | <u>5. Впровадження системи в промислову експлуатацію</u> | |
| <u>1. Призначення та область використання.</u> | <u>6. Наукова новизна.</u> | | | | | | | | | | |
| <u>2. Перегляд аналогічних існуючих систем.</u> | <u>7. Економічна ефективність розробленої програми.</u> | | | | | | | | | | |
| <u>3. Опис і обґрунтування проектних рішень.</u> | <u>8. Заходи з охорони праці та техніки безпеки.</u> | | | | | | | | | | |
| <u>4. Етапи програмування системи.</u> | <u>9. Висновки.</u> | | | | | | | | | | |
| <u>5. Впровадження системи в промислову експлуатацію</u> | | | | | | | | | | | |
| 6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) | | | | | | | | | | | |
| <u>Наукова новизна</u> | <u>1 аркуш</u> | | | | | | | | | | |
| <u>Структурна схема системи</u> | <u>1 аркуш</u> | | | | | | | | | | |
| <u>Функціональна схема системи</u> | <u>1 аркуш</u> | | | | | | | | | | |
| <u>Діаграма процесів</u> | <u>1 аркуш</u> | | | | | | | | | | |
| <u>Блок-схема алгоритму роботи додатку</u> | <u>2 аркуша</u> | | | | | | | | | | |
| <u>Показники економічної ефективності</u> | <u>1 аркуш</u> | | | | | | | | | | |

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

Дудченко А.В. Дослідження та програмна реалізація системи менеджменту файлів ОС Windows 11/Windows Server 2022. 122 Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2023.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи менеджменту файлів ОС Windows 11/Windows Server 2022.

Метою розробки є дослідження та програмна реалізація системи менеджменту файлів ОС Windows 11/Windows Server 2022.

Об'єктом дослідження є процес менеджменту файлів ОС Windows 11/Windows Server 2022.

Предметом дослідження є методи менеджменту файлів ОС Windows 11/Windows Server 2022.

Методи дослідження базуються на методах файлових систем, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи менеджменту файлів ОС Windows 11/Windows Server 2022.

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 11/Windows Server 2022.

Програму розроблено в середовищі Delphi 10.4.

Ключові слова: комп'ютерні науки, менеджмент файлів, ОС Windows 11/Windows Server 2022

ABSTRACT

Dudchenko A.V. Research and software implementation of the OS Windows 11/Windows Server 2022 file management system. 122 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2023.

In this final qualification work for the second (master's) level of higher education, software is developed, which is intended for the OS Windows 11/Windows Server 2022 file management system.

The goal of the development is the research and software implementation of the OS Windows 11/Windows Server 2022 file management system.

The object of the study is the OS Windows 11/Windows Server 2022 file management process.

The subject of the study is the management methods of OS Windows 11/Windows Server 2022 files.

Research methods are based on file system methods, mathematical statistics methods, and software development methods.

The result of the work is the software implementation of the OS Windows 11/Windows Server 2022 file management system.

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 11/Windows Server 2022 OS.

The program was developed in the Delphi 10.4 environment.

Keywords: computer science, file management, OS Windows 11/Windows Server 2022

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	7
1.1 Призначення системи.....	7
1.2 Область застосування.....	8
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	9
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	9
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	21
2.3 Розгорнута постановка завдання	26
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	28
3.1 Опис функціонування системи	28
3.2 Розробка структурної схеми.....	34
3.3 Розробка функціональної схеми	41
3.4 Розробка діаграми процесів.....	43
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	46
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	46
4.2 Захист розробленого програмного забезпечення.....	55
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	59
6 НАУКОВА НОВИЗНА	62

					БКРМ-122.23.0006.00.00.ПЗ			
Вим.	Арк.	№ докум.	Підп.	Дата				
Розроб.	Дудченко А.В.				Дослідження та програмна реалізація системи менеджменту файлів ОС Windows 11/Windows Server 2022	Літ.	Аркуш	Аркушів
Перев.	Доренський О.П.					М	1	103
Н.контр.	Коваленко А.С.				ЦНТУ КН-22М-1			
Затв.	Смірнов О.А.							

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	63
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	63
7.2 Розрахунок трудомісткості розробки програмної продукції.....	65
7.3 Визначення чисельності виконавців і планового фонду зарплати.....	67
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника.....	72
7.5 Визначення собівартості розробки та ціни програмної продукції.....	77
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	79
7.7 Визначення експлуатаційних витрат.....	79
7.8 Визначення економічної ефективності програмної продукції.....	81
7.9 Висновок.....	83
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	84
8.1 Вступ.....	84
8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером.....	85
8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста ...	86
8.4 Розробка заходів з умов поліпшення охорони праці.....	89
8.5 Розрахункова частина	90
8.6 Висновки до розділу.....	94
9 ОСНОВНІ ВИСНОВКИ.....	95
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	97

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

FTP	–	File Transfer Protocol
HDD	–	Hard Disk Drive, жорсткий диск
NFS	–	Network File System
ОС	–	Операційна система

КБПЗ_2023

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

Актуальність теми. Враження від графічного інтерфейсу операційної системи багато в чому визначається зручністю файлового менеджера. Файловий менеджер – комп'ютерна програма, що надає інтерфейс користувача для роботи з файловою системою й файлами. Файловий менеджер дозволяє виконувати найбільш часті операції над файлами – створення, відкриття/програвання/перегляд, редагування, переміщення, перейменування, копіювання, видалення, зміна атрибутів і властивостей, пошук файлів і призначення прав. Крім основних функцій, багато файлових менеджерів включають ряд додаткових можливостей, наприклад, таких як робота з мережею (через FTP, NFS і т.п.), резервне копіювання, керування принтерами та ін.

Виділяють різні типи файлових менеджерів, наприклад:

- Навігаційні і просторові – іноді підтримується перемикання між цими режимами.
- Двопанельні – у загальному випадку мають дві рівноцінних панелі для списку файлів, дерева каталогів і т.п.

Саме з його допомогою виконується пошук, копіювання й переміщення, перейменування й видалення файлів, доступ до архівів. «Провідник», як і всякий стандартний засіб, забезпечує лише необхідний мінімум функцій, даючи можливість навігації по дереву каталогів і доступу до окремих об'єктів за допомогою миші.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи менеджменту файлів ОС Windows 11/Windows Server 2022.

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

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

– Огляд існуючих систем менеджменту файлів ОС Windows 11/Windows Server 2022.

– Дослідження системи менеджменту файлів ОС Windows 11/Windows Server 2022.

– Програмна реалізація системи менеджменту файлів ОС Windows 11/Windows Server 2022.

Об'єктом дослідження є процес менеджменту файлів ОС Windows 11/Windows Server 2022.

Предметом дослідження є методи менеджменту файлів ОС Windows 11/Windows Server 2022.

Методи дослідження базуються на методах файлових систем, методах математичної статистики, методах розробки програмного забезпечення.

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

– Удосконалено метод менеджменту файлів ОС Windows 11/Windows Server 2022.

– Розроблено вітчизняний продукт менеджменту файлів ОС Windows 11/Windows Server 2022, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі менеджменту файлів ОС Windows 11/Windows Server 2022.

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

Робота апробована на LVII Науково-технічній конференції здобувачів вищої освіти «Наука – виробництву», 2023, основні положення випускної

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

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 менеджменту файлів ОС Windows 11/Windows Server 2022, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

КБПЗ_2023

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Основне призначення файлових менеджерів, безумовно, полягає в керуванні файлами. Довгі роки постійного розвитку й удосконалювання подібних програм стали причиною їхнього небувалого розростання вшир. Якщо виразитися іншими словами, то сучасні файлові менеджери містять величезну кількість сервісних функцій, лише побічно пов'язаних з керуванням файлами. І це не дивно. Потенційного покупця необхідно чимсь дивувати, зачаровувати, змушувати звернути увагу саме на цей, конкретний програмний продукт. Звідси росте складність освоєння сучасних файлових менеджерів, системні вимоги. Виходить зворотна ситуація. Комп'ютер прекрасно працює й у руках користувача, нічого не розуміючого у файлових системах. Як наслідок, відбувається деяке зниження популярності файлових менеджерів. Зрозуміло, не всі програмні продукти даного класу складні й важкі.

Робоче вікно складається із двох файлових панелей, властивій більшості подібних програм. Усередині кожної з них можна переглядати дані в режимі піктограм, списку й у вигляді таблиці. Сортування файлів міняється шляхом клацання лівою кнопкою миші по заголовках колонок. У головному меню подібних опцій немає. Список файлів може бути відфільтрований за деякими критеріями. По-перше, допускається відображення всіх файлів, крім схованих і системних елементів. Другий режим дозволяє вводити розширення вручну. І, нарешті, третій варіант припускає показ тільки виділених файлів. Основні операції з файлами виконуються або за допомогою гарячих клавіш, або шляхом натискання кнопок на панелі інструментів. Усередині кожної кнопки перебуває інформація про те, яка клавіша їй відповідає. Більшість файлових менеджерів мають стандартизовані клавіші керування основними операціями. Перехід між

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

папками може здійснюватися не тільки за допомогою вибору відповідних елементів усередині файлових панелей. Їхні заголовки являють собою рядок уведення адреси. Іноді швидше ввести коротке ім'я папки, ніж довго переміщатися до рядка з її ім'ям. У нижній частині робочого вікна додатка перебуває командний рядок, що повністю дублює її системний аналог.

Вибір логічних розділів здійснюється шляхом вибору відповідної кнопки на панелі інструментів. Якщо її немає, можна скористатися спеціальним меню.

1.2 Область застосування

Файловий менеджер відіграє роль посередника між програмами й даними й користувачем. Незважаючи на гаданий прогрес у цій області, сьогоднішні менеджери додержуються канонів, заявлених програмою Norton Commander. Не випадково для багатьох улюбленою оболонкою залишається файловий менеджер із класичним двовіконим інтерфейсом. Необхідність двох панелей продиктована зручністю виконання двомісних операцій, а наявність додаткових функцій на зразок перегляду текстових і графічних файлів, роботи з FTP-серверами, безпосереднього доступу до архівів розповсюджених форматів забезпечує сучасному файловому менеджеру законне місце в обов'язковому наборі програм.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи менеджменту файлів ОС Windows 11/Windows Server 2022, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

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

Розглянемо існуючі популярні менеджери файлів.

FAR

Популярний менеджер файлів і архівів з текстовим інтерфейсом. Його сильна сторона – модулі, що підключаються.

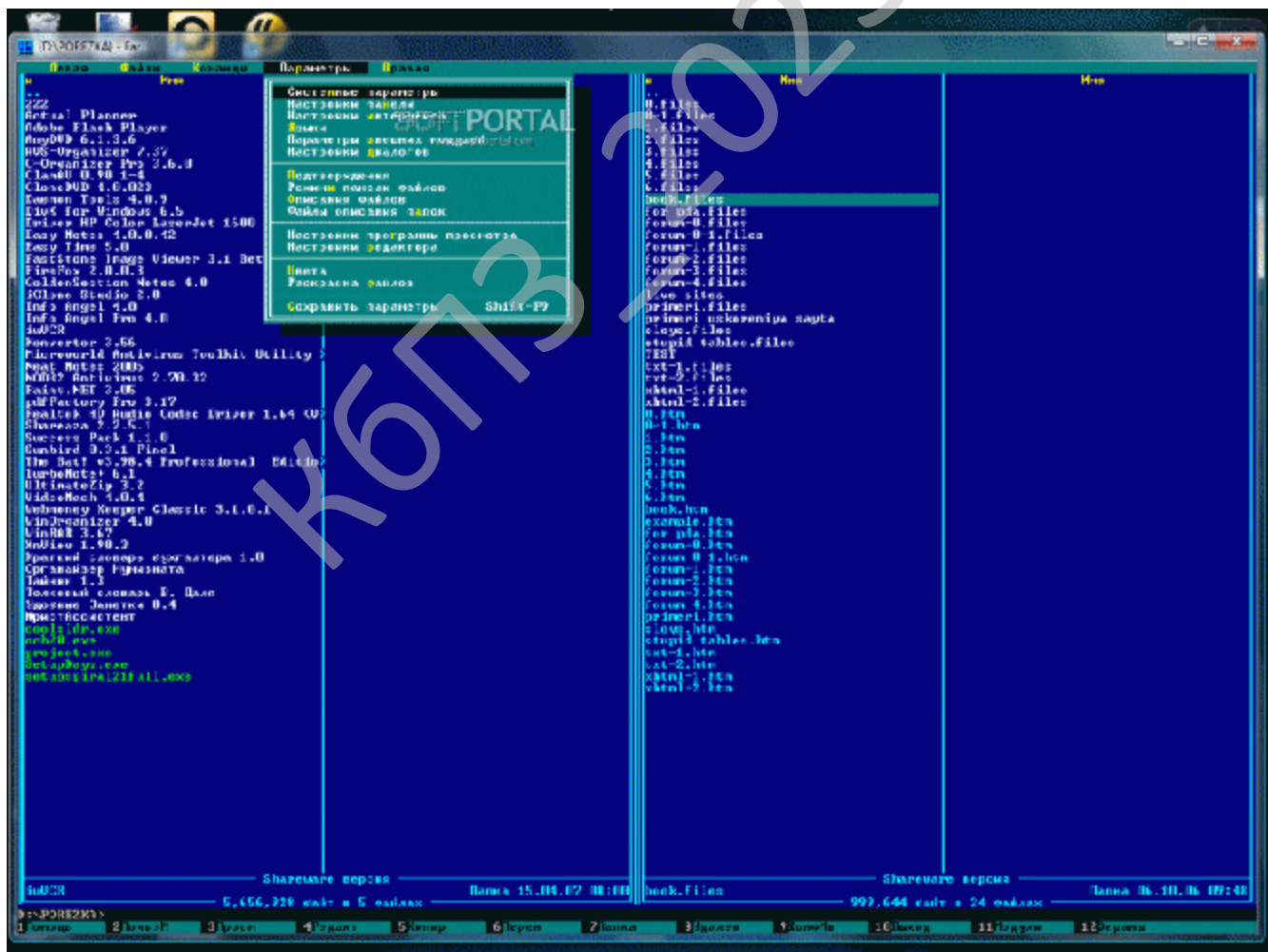


Рисунок 2.1 – Интерфейс користувача FAR

У даний момент доступно кілька сотень модулів, від простих до екзотичних – наприклад, що дозволяють переглядати вміст образів дисків MKDOS-емуляторів БК-0010-01 і БК-0011м. Існують модулі-календарі й калькулятори, ігри, конвертори й засоби перегляду. Завдяки налаштуванню кольорів, безлічі варіантів відображення й сортування, програма забезпечує наочне подання списків файлів у папках. Передбачено порівняння папок, робота із системним буфером обміну, видалення файлів в «Кошик», кілька способів перетворення регістра букв в іменах файлів і робота з архівами. У комплект входить зручний FTP-клієнт. У цілому FAR можна охарактеризувати як простий у використанні, швидким, оснащеним вичерпним набором функцій. Текстовий режим відображення не повинен уводити вас в оману: це повністю 32-розрядна програма, для роботи якої необхідна операційна система Windows 95 або більш пізня.

Frigate

Розроблювач так описує Frigate: «набудовуване й комфортабельне середовище для керування файлами». У дійсності керування файлами – лише невелика частина функцій цієї програми. Підтримуючи популярні формати упакування (включаючи ZIP, ARJ, RAR, ACE і LHA), Frigate забезпечує «прозору» роботу з архівами. Програма оснащена засобом перегляду й перетворення зображень; дозволяє одержати доступ до тегів звукозаписів і інформації про відеозаписи. Передбачено перегляд і редагування документів у форматі TXT, XML, HTML, XLS і DBF.

Для операцій з файлами використовується інтерфейс із двома панелями. Вибір команд продубльований загальноприйнятим набором гарячих клавіш і кнопками панелі інструментів, що набудовується. Менеджер підтримує операції перетаскування, забезпечує доступ до контекстного меню «Провідника», видаляє файли в «Кошик», працює з буфером обміну й виконує команди, введені в командному рядку.

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

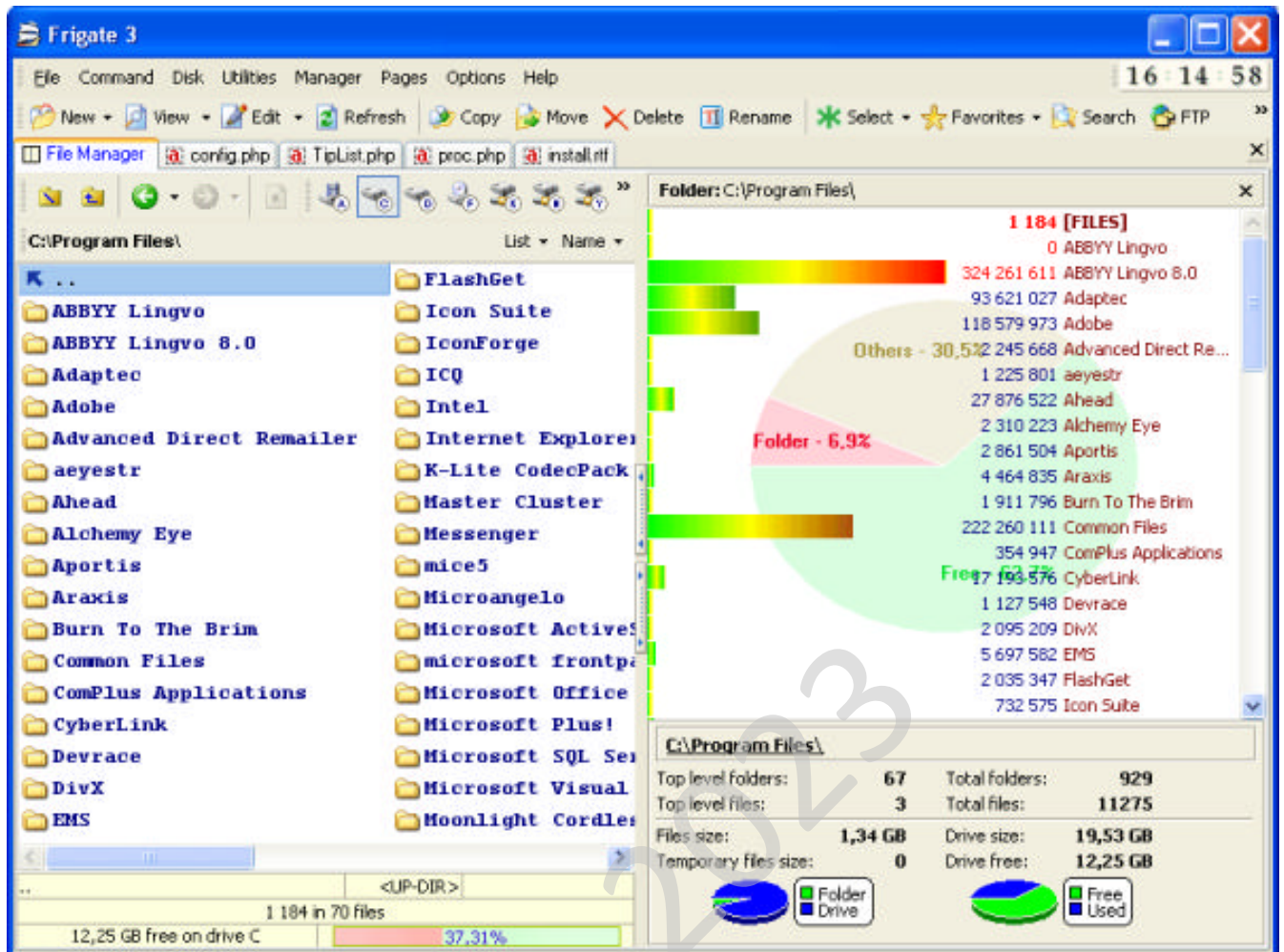


Рисунок 2.2 – Інтерфейс користувача Frigate

Передбачено роботу з FTP. Цікава особливість програми – можливість одночасного виконання декількох команд у паралельних потоках. На рівні модулів, що підключаються, програма сумісна з менеджером Total . Можливості Frigate доповнюють вбудовані утиліти: калькулятор, блокнот, органайзер, телефонна книга, будильник. Програма розрахована на експлуатацію під керуванням Windows 95/98/Me/NT/2000/XP.

Total Commander

Раніше відомий, як Windows Commander, цей менеджер являє собою потужний інструмент для роботи з файлами. Інтерфейс програми відрізняється своєрідністю: можливо, у ньому проглядається вплив колишніх версій Windows.

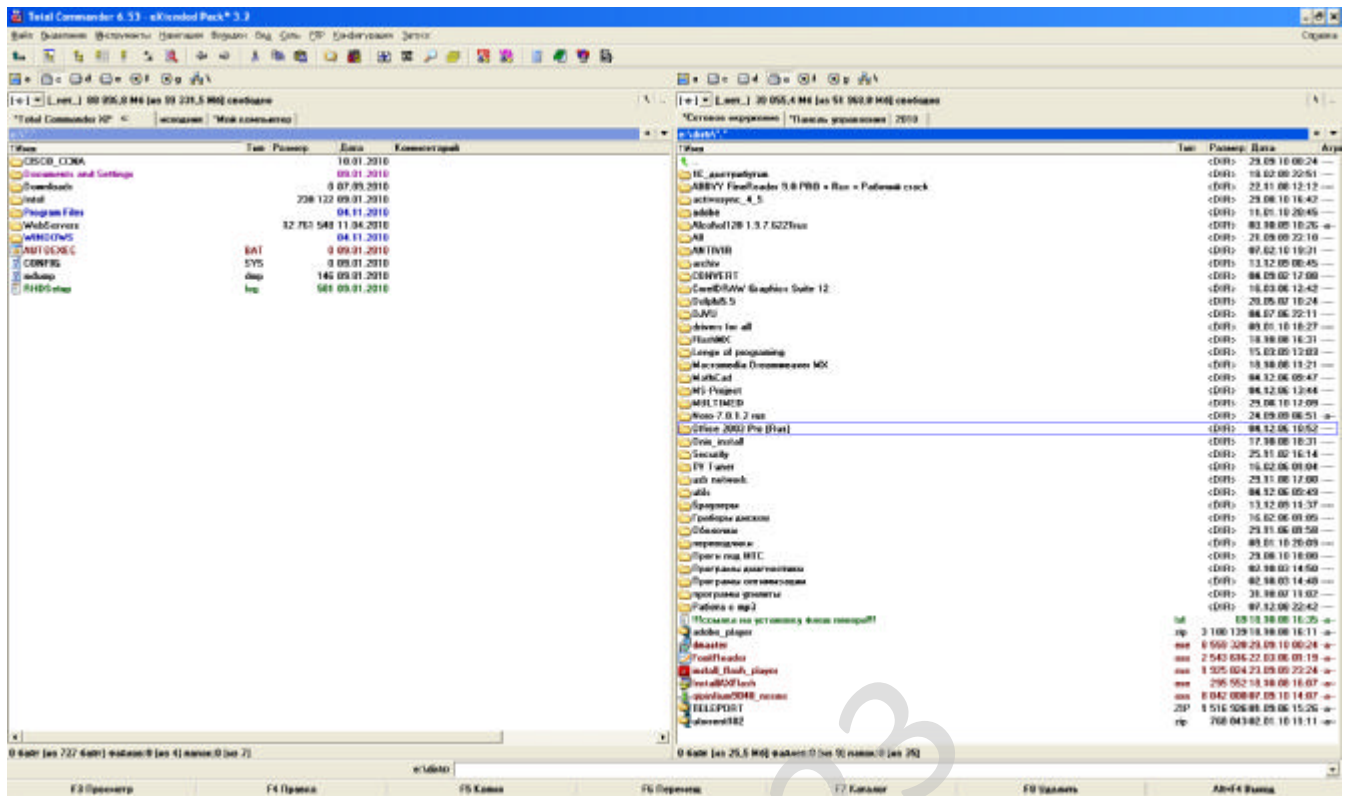


Рисунок 2.3 – Інтерфейс користувача Total Commander

Передбачено видалення в «Кошик», мінімізація в системну область панелі завдань, перегляд зображень і текстів найпоширеніших форматів, відтворення звукозаписів. Серед цікавих особливостей – можливості групового перейменування (з автоматично інкрементуємим суфіксом в ім'ї файлу), поділу файлів на частини, декодування файлів поштових форматів MIME, UUE, XHE, WinHex, порівняння або синхронізації папок. Запуск додатків забезпечують рядок для введення команд, конфігуруєма панель інструментів і меню швидкого запуску. Вбудований FTP-клієнт має можливість поновлення завдань, з'єднання через проксі-сервер і передачі файлів прямо між двома серверами. Реалізовано функції порівняння файлів, синхронізації папок, а також папки й архіву у форматі ZIP, пошуку дублікатів, пошуку по контексту з можливістю «заглянути» в архіви. Робота з архівами у форматах ZIP, ARJ, LZH, TAR, GZ, CAB, RAR і ACE здійснюється, як зі звичайними папками. Великі архіви можна обробляти у фоновому режимі. За допомогою цієї програми й підходящого шнура можна

з'єднати два комп'ютери через паралельні порти. Працює Total Commander під керуванням Windows 10/11/Windows Server 2022.

AB Commander

У минулому ця програма називалася Singularity. Використовуючи знайому ідею інтерфейсу головного вікна із двома панелями, вона повністю замінєє «Провідника» Windows. Одночасна робота із двома папками полегшує виконання багатьох файлових операцій, роблячи їх більше ефективними. AB Commander пропонує багатий набір інструментів для навігації по файловій системі й керування файлами. Можете заощадити час, використовуючи зручну панель пристроїв, списки вибраного й історії, панелі інструментів, що налаштовуються, і гарячі клавіші.

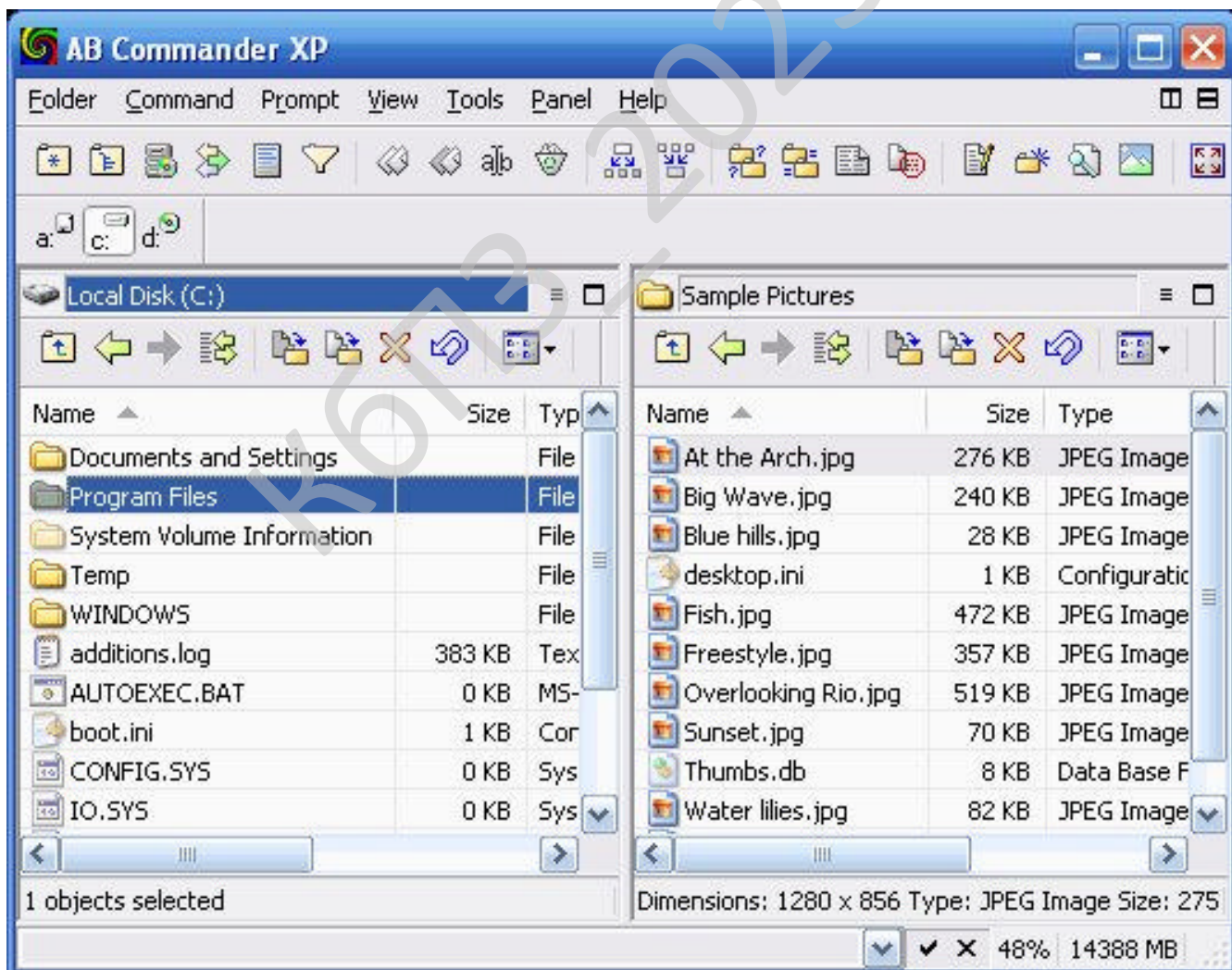


Рисунок 2.4 – Інтерфейс користувача AB Commander

Менеджер, оснащений інтерфейсом для введення команд у командному рядку, дозволяє фільтрувати, порівнювати й синхронізувати вміст папок, забезпечує звертання до звичайним, віртуальним і стислим папкам за допомогою того самого уніфікованого інтерфейсу. Серед додаткових можливостей – поділ великих файлів на частини, перегляд зображень, друк списків файлів. Програма АВ Commander працює на комп'ютерах з ОС Windows 10/11/Windows Server 2022.

FreeCommander

FreeCommander це легка у використанні альтернатива стандартному провідникові Windows. Програма має всі необхідні функції для керування даними. Настановну папку можна скопіювати на CD або USB, щоб працювати із цією програмою на сторонньому комп'ютері.

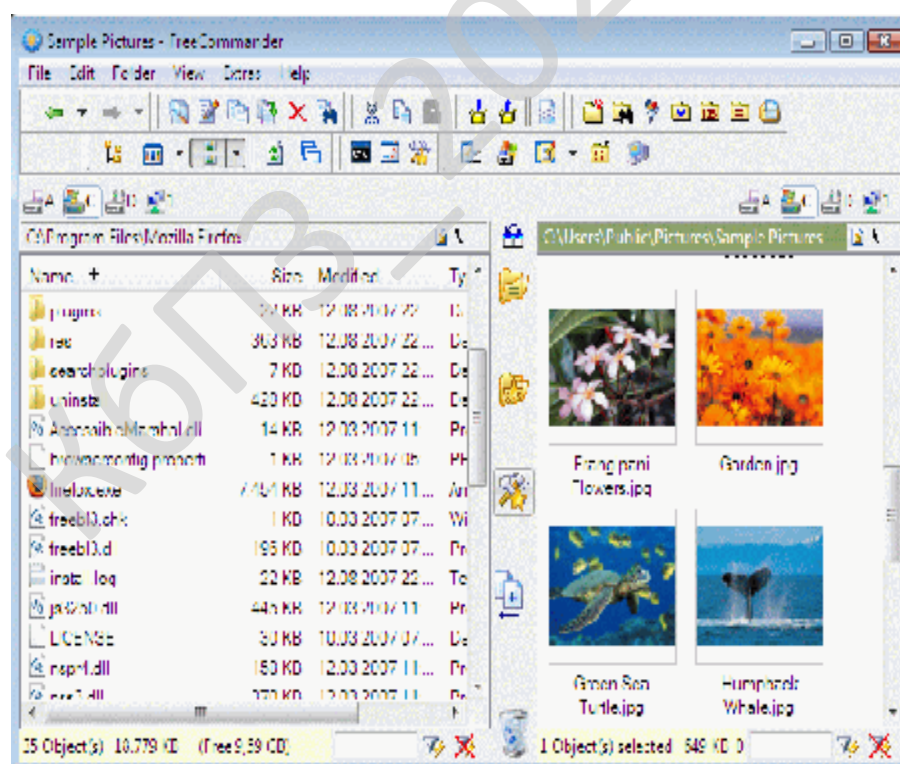


Рисунок 2.5 – Інтерфейс користувача FreeCommander

функцій для роботи з файлами й директоріями на локальному комп'ютері: копіювання, переміщення, видалення, пошук, фільтрація, перегляд по заданих параметрах і т.д. Програма може працювати як з окремо взятими файлами, так і з файлами в пакетному режимі.

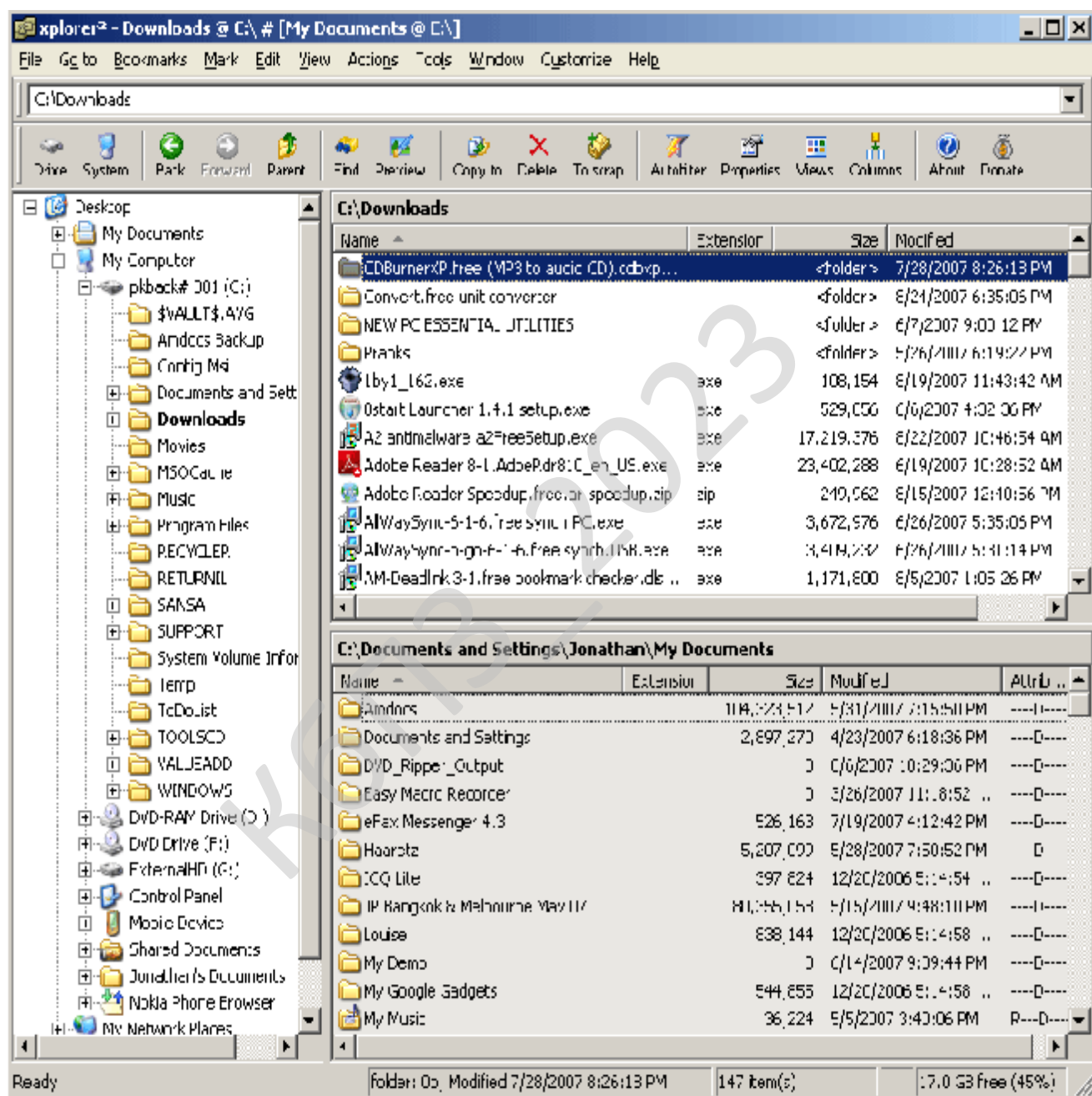


Рисунок 2.6 – Інтерфейс користувача Xplorer2 Lite

Серед інших можливостей цього файлового менеджера – робота з мережними дисками. У підключених мережних дисках при необхідності відбувається автоматичне відновлення файлів.

Основні можливості Xplorer2 Lite:

- Повний доступ до простору імен командного рядка.
- Автоматичне відстеження змін у всіх папках (включаючи мережні).
- Багатопоточна й багатовіконна обробка.
- Обробка двох папок в одному вікні.
- Інтерфейс із підтримкою вкладок.
- Комбінації клавіш швидкого виклику.
- Автоматичне виконання зазначеного шляху.
- Фільтр видимості об'єктів.
- Збереження й керування інтерфейсом в Windows.
- Експорт інформації про папку у вигляді тексту (напр. для друку).
- Тверді посилання (NTFS).
- Пакетне перейменування.
- Знаходження тексту у файлах.
- Пошук по ключових словах з логічним контекстом.
- Підтримка Unicode.
- Розбивка/Об'єднання/Видалення.
- Редактор тексту.
- Попередній перегляд Text/RTF/Hex.
- Режим перегляду ескізів.
- Порівняння й синхронізація папок.
- Налаштування панелі інструментів.
- Об'єднання папок, символні посилання (NTFS).
- Підрахунок обсягу папок у детальному перегляді.
- Налаштування зовнішнього вигляду панелі інструментів (скіни).

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

- Групове перейменування файлів і каталогів
- Синхронізація каталогів
- Підтримка архівів ZIP, RAR, ACE, CAB, JAR, TAR, LHA, GZ, TGZ, ARJ
- Вбудований FTP-клієнт
- Таби каталогів
- Підтримка WLX-плагінів і WCX-плагінів
- Вбудований проглядач і функція швидкого перегляду
- Робота з мережним оточенням
- Підтримка Drag & Drop при роботі з іншими додатками
- Кнопки історії й Вибране
- Фонове копіювання/перенос/видалення
- Безпечне видалення файлів (WIPE)
- Використання фонових рисунків
- Стили оформлення: колірні категорії файлів, шрифти для всіх елементів інтерфейсу
- і інше.

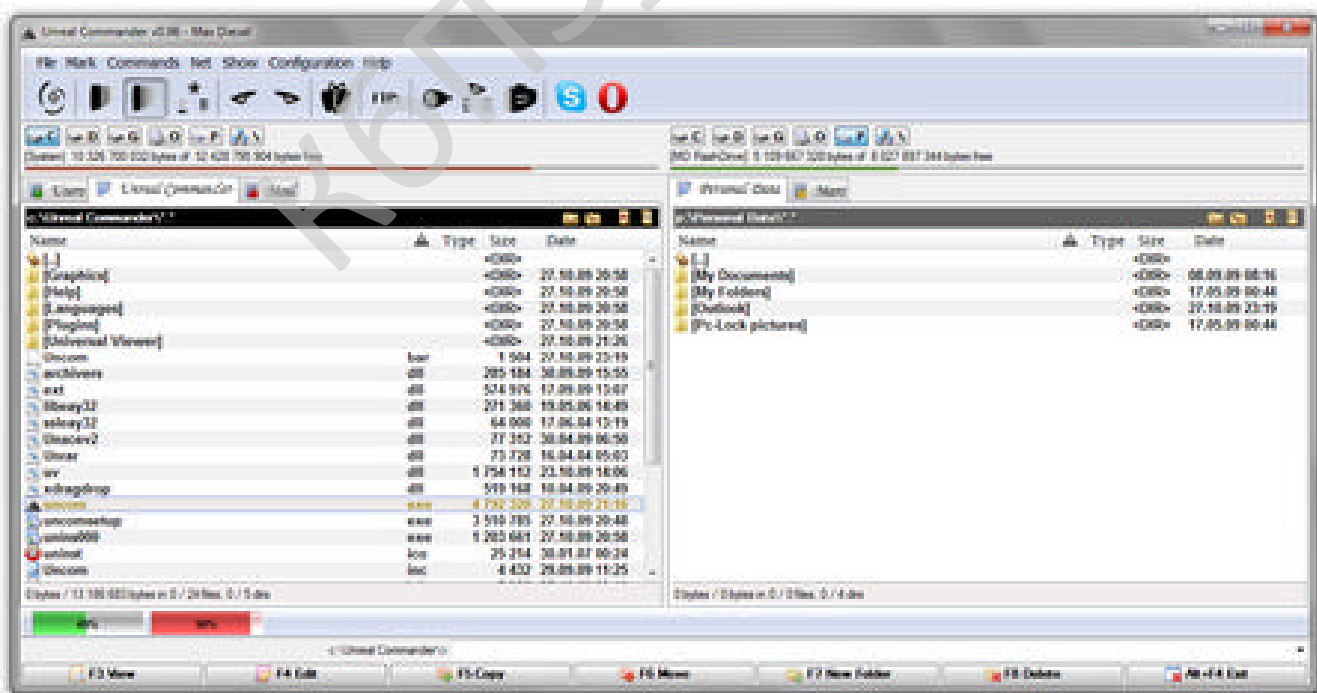


Рисунок 2.7 – Інтерфейс користувача Unreal Commander

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.

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.

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

– Тип даних Delphi «record» тепер підтримуть 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.

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

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

– Додані оновлені драйвери для FireBird, PostgreSQL і SQLite.

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

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

RAD Studio 10.4 Короткий огляд:

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

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

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

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

Істотне поліпшення Delphi Code Insight

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

Delphi Custom Managed Records

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

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

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 ІМЕ.
- 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.

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

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

забезпечення, яке призначено для системи менеджменту файлів ОС Windows 11/Windows Server 2022.

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

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

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

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

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

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

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

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

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

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Програмний продукт, який розробляється у результаті виконання магістерського проектування, дозволяє реалізувати наступні можливості:

- Двопанельний багатомовний графічний інтерфейс користувача.
- Сполучення клавіш, що налаштовуються повністю, близькі файловому менеджеру Norton Commander для DOS (клавіатурні комбінації для перегляду, редагування, копіювання й видалення файлів).
- Підтримка Drag&Drop за допомогою миші.
- Вкладочний інтерфейс.
- Панелі меню, що налаштовуються й, інструментів із кнопками для запуску зовнішніх програм або внутрішніх команд меню.
- Виділення й підсвічування файлів на панелі по імені, розширенню, даті й часу створення файлу й по багатьом інших умовам.
- Вбудований FTP-клієнт, з підтримкою SSL/TLS.
- Робота з архівами так само, як з підкаталогами. Відповідна програма-архіватор викликається автоматично. Підтримуються PKZIP, ARJ, LHA, RAR, UC2, ACE, TAR, gzip і tar+gzip.
- Внутрішній пакувальник для формату ZIP, з можливістю створення багатотомних і що саморозпаковуються (16– і 32-бітних) архівів.
- Внутрішні розпаковщики для форматів ZIP, ARJ, LZH, gzip, tar, RAR і ACE.
- Розрізування й склеювання файлів великого обсягу.
- Підрахунок і перевірка контрольних сум файлів (CRC32, MD5, SHA1).
- Групове перейменування файлів і синхронізація каталогів.

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

- Порівняння файлів з можливістю редагування, порівняння файлів з різними кодовими сторінками.
- Розширений пошук файлів, включаючи пошук тексту в будь-яких файлах, у тому числі на декількох дисках і FTP, пошук дублікатів файлів, пошук усередині архівів.
- Вбудований переглядач файлів (lister) для перегляду файлів будь-якого розміру в шістнадцятковому, двійковому або текстовому форматі з використанням набору символів «DOS», «ANSI»/«Windows», Unicode, UTF-8 і інших.
- Підтримуються регулярні вираження при пошуку файлів (в іменах і у вмісті) і у внутрішньому переглядачі, груповому перейменуванні файлів, діалозі вибору й виділення файлів і фільтрах.
- Підтримка послідовної черги й паралельного фонового виконання операцій над файлами (копіювання, перенос, видалення, робота з архівами й FTP).
- Журналювання файлових операцій.
- Можливість тимчасового підвищення рівня привілеїв. підтримка механізму User Account Control в Windows Vista.
- Природна (native) підтримка портативного режиму роботи (без інсталяції), підтримка U3 флеш-накопичувачів.
- Розширення функціональності програми за допомогою модулів, що підключаються, (плагінів).
- Працює з довгими (більше 255 символів) шляхами під NTFS.
- Безліч операцій тепер підтримують фоновий режим:
 - а) упакування й розпакування різних видів стандартних архівів;
 - б) упакування й розпакування з використанням плагінів (необхідні доробка плагіна);
 - в) копіювання в/з плагінів файлової системи (необхідна доробка плагіна);
 - г) підрахунок розміру вмісту підкаталогів по Alt+Shift+Enter і пробілі;

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

- д) підрахунок і перевірка CRC-сум;
- е) розцвічення файлів при використанні полів WDX-плагінів з фоновим підрахунком;
- ж) одержання міток диска при відкритті списку по Alt+F1/F2.
 - Поліпшення в інструменті синхронізації:
 - а) реалізовано порівняння деяких архівів по вмісту (ARJ, LHA, TAR, RAR);
 - б) розширено інтерфейс WDX-плагінів для власної реалізації порівняння файлів;
 - в) додано можливість ігнорувати сховані/системні файли, а також символічні посилання;
 - г) підтримка порівняння по CRC-сумах для FTP, якщо сервер підтримує таку функціональність (опція "по вмісту").
 - Додано підтримку USB-USB підключень двох комп'ютерів з використанням кабелів, сумісних зі специфікацією Easy Transfer.
 - Режим списку файлів без підкаталогів для виділених об'єктів (Ctrl+Shift+B).
 - Поліпшення в роботі FS-плагінів тимчасової панелі.
 - Додано підтримку переспрямованих локальних дисків у термінальній сесії (\tsclient).
 - Підтримка одержання списку файлів з FTP командою MLSD (більше точні часові мітки, незалежність від локального часу, стандартизований формат списку).
 - Частина, що розкривається, з опціями в діалозі копіювання тепер можна тримати завжди розкритою.
 - Збільшено максимальну довжину рядка в Lister'e до 1024 символів.
 - Для RAR-архівів із зашифрованим заголовком тепер як вміст відображається псевдо-файл із ім'ям "Зашифроване -> F5" (щоб відрізнити від порожнього архіву).

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

- Можливість зберігати стан опції розпакування архівів в окремі підкаталоги.
- В 64-бітних системах відображаються окремі підкаталоги, що неперенаправлялися, system32 (catroot, drivers\etc і т.п.), які не видні через загальний перенапрямок в SysWOW64.
- Розширено контекстне меню в результатах діалогу пошуку (додане 64-бітне підменю й внутрішні асоціації).
- Список вибраного можна викликати подвійним клацанням по панелі адреси, не активуючи перехід по ланцюжку: досить утримувати натиснутим Ctrl.
- Для парних і непарних рядків у файлової панелі можна використовувати різний колір тла.
- Інструмент порівняння файлів містить вбудований редактор.
- Додаткова панель дерева папок. Може бути використана як загальна для двох панелей, так і для кожної панелі окремо.
- Вбудований переглядач (можливість центрувати зображення, текстовий курсор, можливість змінювати розмір тільки для більших зображень).
- Вікно зміни атрибутів файлів і папок підтримує контентні плагіни.
- Можливість ведення журналу всіх файлових операцій.
- Відображення букви диска у вкладках (опціонально).
- Користувальницькі колонки й режим мініатюр для системних плагінів.
- У діалоговому вікні копіювання-перезапису можливі додаткові операції: порівняння по вмісту, перейменування файлу, автоматичне перейменування, передпрогляд зображень, і т.д.
- Сортування по декількох колонках (Ctrl+клік по заголовку додаткової колонки).
- Підключення до FTP через SSL/TLS (необхідно скачати й установити додаткові .dll файли від стороннього виробника).
- Використання користувальницьких команд у головному меню й для комбінацій клавіш.

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

Плагіни

Функціональність Файлового менеджера можна розширювати за допомогою плагінів.

Архіваторні плагіни (Packer Plugins, WCX)

Архіваторні плагіни використовуються для розпакування певних типів файлів, звичайно архівних форматів. Деякі такі плагіни також дозволяють створювати нові архіви підтримуваного типу й модифікувати існуючі архіви.

Плагіни можуть використовуватися для:

- Упакування/розпакування архівів bzip2, rpm, HA і інших архівів підвищеного рівня компресії.
- Створення списку файлів в обраних каталогах, що корисно для каталогізації цілих дисків.
- Створення командного файлу для обраних файлів, наприклад, для кількаразового пакетного копіювання.
- Розпаковування, пошуку, розпакування й видалення файлів в СНМ-файлах.
- Перегляду й добування ресурсів (музики, текстур і т.д.) з ігрових архівних файлів.
- Приховання файлів на флеш-диску або дискеті від сторонніх очей.
- Симетричного й асиметричного шифрування й дешифрування файлів симетричними й асиметричними алгоритмами.
- Конвертації звукових, графічних файлів.
- Стеганографічного приховання файлів у зображенні.

Плагіни внутрішньої програми перегляду (Lister Plugins, WLX)

Плагіни внутрішньої програми перегляду відображають спеціальні типи файлів в вбудованому переглядачі (Lister), а також на панелі швидкого перегляду.

Приміром, плагіни можуть використовуватися для:

- Відображення файлів у графічних форматах, баз даних, PDF, openoffice.org.

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

– Відображення вихідного коду на C, Pascal, Java і т.д. З підсвічуванням синтаксису.

– Програвання файлів у звукових– і відеоформатах, а також CD-треків.

– Перегляду різної інформації про архіви: розмір даних, кількість файлів і т.д.

– Перегляду SWF-файлів, веб-страниц, електронних книг формату FB2.

Плагіни файлової системи (File System Plugins, WFX)

Плагіни файлової системи доступні через панель Мережного оточення. Вони звичайно забезпечують доступ до якихось частин комп'ютера, куди немає доступу через імена дисків, або до яких-небудь віддалених систем.

Плагін файлової системи може бути призначений для доступу до:

– Файлової системі мобільних телефонів.

– Файловим системам ext2fs і reiserfs.

– Віддалених веб-серверів по HTTP-протоколу, поштовому серверу.

– Інформації про запущені на даний момент процесах.

– Вмісту кешу Internet Explorer.

– Базі (контактам і історії повідомлень) Міранди.

– Вмісту плейлистів Winamp.

– Реєстру Windows, у тому числі на віддаленому комп'ютері, з можливістю роботи із ключами реєстру як з файлами й папками.

– Буферу обміну Windows, наприклад, копіюванню графічного фрагмента в різних форматах.

– Схованих налаштувань операційної системи.

Інформаційні (контентні) плагіни (Content plugins, WDX)

Інформаційні плагіни покликані вирішувати кілька завдань: пошук по специфічних властивостях і відомостям про вміст файлів, відображення цієї інформації у файлових списках і використання її в інструменті групового перейменування й для додавання додаткової інформації в назву файлу.

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

Контентні плагіни можуть застосовуватися для одержання:

- Тегів аудіофайлів (виконавець, заголовок, альбом і т.д.).
- Інформації про відеофайли, графічні файли, цифрові фотознімки, що зберігається в заголовках exif в jpeg-файлах.
- Тексту з документах ms word, openoffice.org, acrobat.
- Атрибутів файлу, таких як час створення й т.п. (вбудований плагін).
- Атрибутів модулів, що виконуються.

3.2 Розробка структурної схеми

Структурна схема складається з наступних блоків:

- Блок файлових панелей.
- Блок меню користувача.
- Блок швидкого доступу до елементів меню.
- Блок гарячих клавіш.

Блок файлових панелей включає до себе наступні структурні елементи:

- Ім'я файлу.
- Тип файлу.
- Розмір файлу.
- Дата створення файлу.
- Атрибути файлу.
- Коментар.
- Блок вибору диску, з яким необхідно працювати.
- Закладки на каталоги, до яких найчастіше звертаються.
- Дерево файлів.
- Мережне оточення.
- Менеджер плагінів.

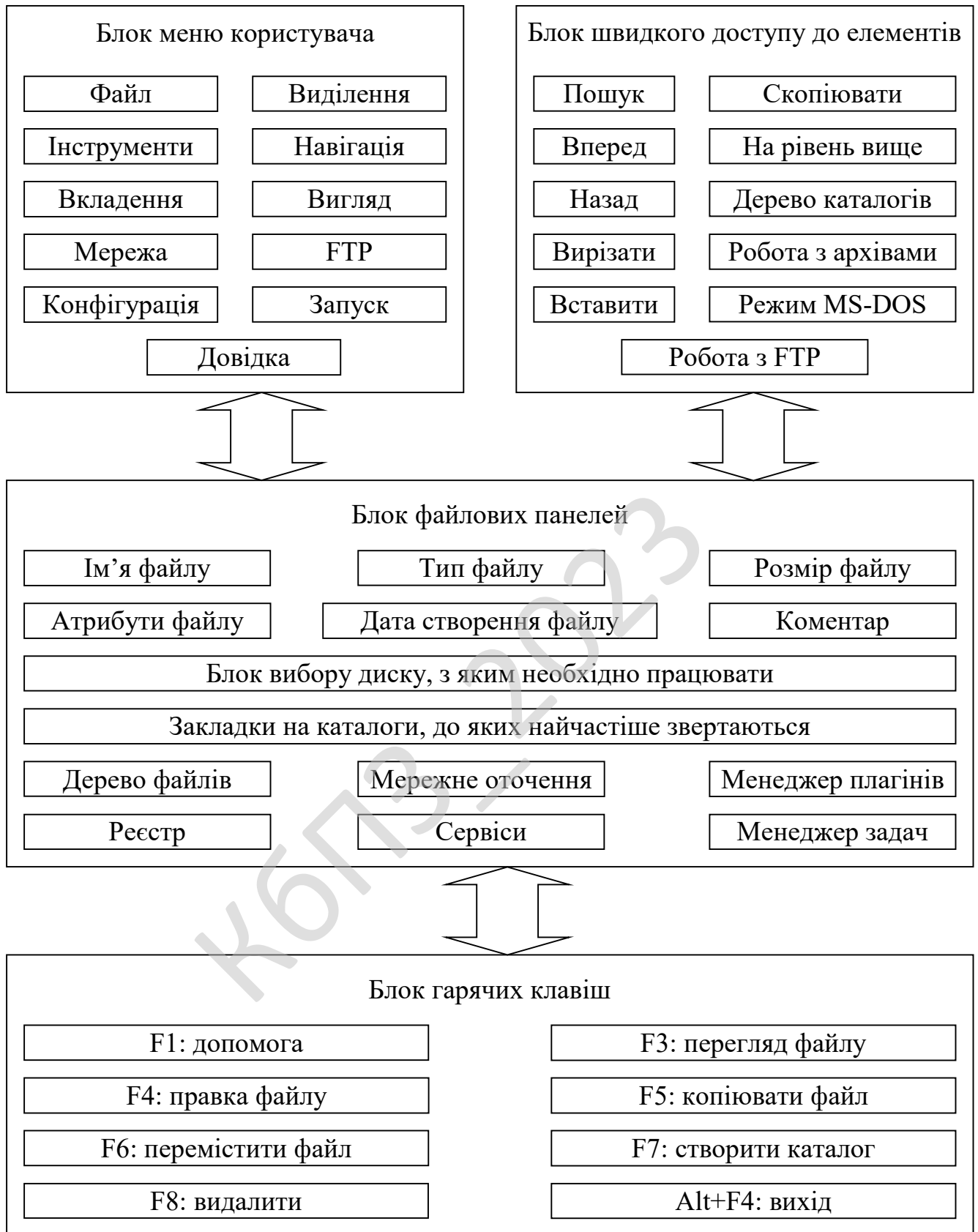


Рисунок 3.1 – Структурна схема системи

- Реєстр.
- Сервіси.
- Менеджер задач.

Блок меню користувача включає в себе наступні елементи:

- Файл:
 - а) запуск файлу під курсором;
 - б) запуск файлу від імені;
 - в) відкрити за допомогою;
 - г) змінити атрибути;
 - д) коментар;
 - є) об'єм файлу;
 - ж) контекстне меню;
 - з) поміняти за змістом;
 - і) друк;
 - к) упакувати;
 - л) розпакувати;
 - м) протестувати архіви;
 - н) створити ярлик;
 - о) розбити файл;
 - п) зібрати файл;
 - р) кодувати;
 - с) декодувати;
 - т) підрахувати СРС-суму;
 - у) вихід.
- Виділення:
 - а) виділити групу;
 - б) зняти виділення;
 - в) виділити все;
 - г) зняти виділення усього;

- д) виділити файли за маскою;
- е) зняти виділення за маскою;
- ж) інвертувати виділення;
- з) зберегти виділення;
- і) відновити виділення;
- к) порівняти каталоги;
- л) відмітити нові, заховати старі.

– Інструменти:

- а) інформація про систему;
- б) мітка диску;
- в) пошук файлів;
- г) групове перейменування;
- д) синхронізувати каталоги;
- е) копіювати обрані файли у буфер;
- ж) вирізати обрані файли;
- з) вставити обрані файли;
- і) копіювати путь в командний рядок;
- к) очистити командний рядок;
- л) завантажити сеанс DOS.

– Навігація:

- а) ліва панель;
- б) права панель;
- в) робочий стіл;
- г) мій комп'ютер;
- д) панель управління;
- е) шрифти;
- ж) мережне оточення;
- з) принтери;
- і) назад;

- к) вперед;
- л) відкрити каталог/архів;
- м) історія каталогів;
- н) дерево каталогів;
- о) обрані каталоги;
- п) перехід до наступного диску;
- р) перехід до попереднього диску;
- с) оновити зміст панелі;
- т) попередній командний рядок;
- у) наступний командний рядок.

– Вкладення:

- а) показати меню вкладення;
- б) нове вкладення для поточного файлу;
- в) блокування/розблокування вкладення;
- г) відкрити папку у новому вкладенні;
- д) перехід на наступне вкладення;
- є) перехід на попереднє вкладення;
- ж) закрити вкладення;
- з) закрити усі вкладення.

– Вигляд:

- а) розширені налаштування;
- б) стислий;
- в) докладний;
- г) коментарі;
- д) користувальницькій набір колонок;
- є) дерево каталогів;
- ж) швидкий перегляд;
- з) сортування;
- і) фільтр;

- к) розташувати панелі одна над одною;
- л) показати розміри усіх папок.
- Мережа:
 - а) підключити мережний диск;
 - б) відключити мережний диск;
 - в) зробити поточний каталог загальним;
 - г) закрити загальний доступ до каталогу;
 - д) зміна прав доступу;
 - є) аудит файлів;
 - ж) стати володарем;
 - з) показати ресурси адміністратора.
- FTP:
 - а) з'єднатися з FTP-сервером;
 - б) нове FTP-з'єднання
 - в) роз'єднатися з FTP-сервером;
 - г) додати у список завантаження;
 - д) диспетчер фонові пересилки;
 - є) завантаження за списком;
 - ж) показувати відкриті файли на сервері;
 - з) обрати режим передачі (авто/двійковий/текст).
- Конфігурація:
 - а) налаштування;
 - б) панель інструментів;
 - в) налаштування: вигляд;
 - г) налаштування: екран;
 - д) налаштування:
 - є) налаштування: кольор;
 - ж) налаштування: табулятори;
 - з) налаштування: мова;

- і) налаштування: операції;
 - к) налаштування: правка/перегляд;
 - л) налаштування: архіватори.
 - м) налаштування: різне;
 - н) редагувати файли конфігурації;
 - о) запам'ятати налаштування.
- Запуск:
- а) клавіатурні команди;
 - б) сайт підтримки;
 - в) інструменти;
 - г) список файлів/папок;
 - д) запустити програму з більш високим пріоритетом;
 - є) створити віртуальний диск;
 - ж) відключити віртуальний диск;
- Довідка:
- а) зміст;
 - б) гарячі клавіші;
 - в) сайт програми;
 - г) про програму.

Блок швидкого доступу до елементів меню включає в себе наступні команди:

- на рівень вище;
- дерево каталогів;
- пошук;
- вперед;
- назад;
- вирізати;
- скопіювати;
- вставити;

- режим MS-DOS;
- робота з архівами;
- робота з FTP.

Блок гарячих клавіш:

- F1: допомога.
- F3: перегляд файлу.
- F4: правка файлу;
- F5: копіювати файл;
- F6: перемістити файл;
- F7: створити каталог;
- F8: видалити;
- Alt+F4: вихід.

3.3 Розробка функціональної схеми

На рисунку 3.2 зображена функціональна схема системи. Нижче розглянемо її більш докладно. Вона складається з наступних функціональних блоків:

- Двопанельна технологія відображення файлів – горизонтально й вертикально.
- Закладний інтерфейс.
- Опціональне дерево папок для кожної панелі.
- Вбудований файловий переглядач для перегляду файлів у шістнадцятковому, двійковому, текстовому або у форматі зображення.
- Переглядач файлів працює й усередині архівів.
- Вбудоване керування архівами: ZIP (читання, запис), CAB (читання, запис), RAR (читання).
- Вбудований FTP клієнт.

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41



Рисунок 3.2 – Функціональна схема системи

– Легкий доступ до системних папок, панелі керування, робочого столу й меню Пуск.

- Копіювання, переміщення, видалення, перейменування файлів і папок.
- Інструмент мульти-перейменування.
- Затирання файлів.
- Створення й перевірка контрольних сум MD5.
- Поділ файлів.
- Властивості файлів і контекстне меню.
- Обчислення розміру папки.
- Порівняння / синхронізація папок.
- Зміна дати й атрибутів файлу.
- Вибрані папки / програми.
- Пошук файлів (також усередині архівів).
- Файлові фільтри для відображення.
- Задані користувачем колонки для докладного перегляду.
- Командний рядок DOS.
- Мультимовна підтримка.

Крім того є ще блок плагінів:

- Архіваторні плагіни (Packer Plugins, WCX).
- Плагіни внутрішньої програми перегляду (Lister Plugins, WLX).
- Плагіни файлової системи (File System Plugins, WFX).
- Інформаційні (контентні) плагіни (Content plugins, WDX).

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

3.4 Розробка діаграми процесів

Діаграма взаємодії процесів системи, розробленої у результаті виконання магістерського проектування, наведена на рисунку 3.3.

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

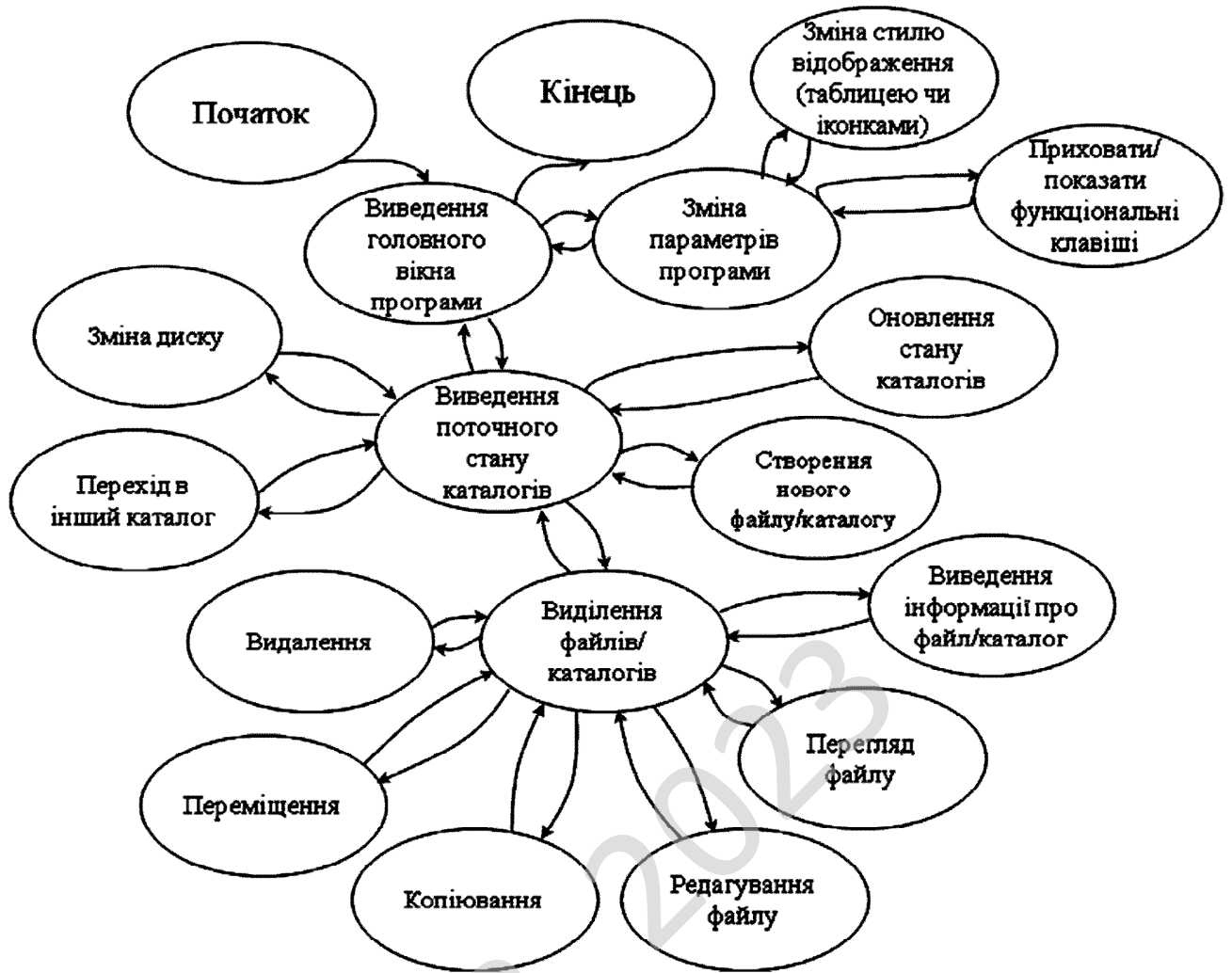


Рисунок 3.3 – Діаграма процесів системи

З діаграми ми бачимо, що першим процесом, який запускається у системі, є процес виведення головного вікна програми.

Цей процес взаємодіє з наступними процесами:

- Процес виведення поточного стану каталогів.
- Процес зміни параметрів програми.

Процес виведення поточного стану каталогів, у свою чергу, взаємодіє з наступними процесами:

- Процес заміни диску.
- Процес переходу у інший каталог.
- Процес видалення.

- Процес оновлення стану каталогів.
- Процес створення нового файлу/каталогу.
- Процес виділення файлів/каталогів.

Останній процес взаємодіє з наступними процесами:

- Процес видалення.
- Процес переміщення.
- Процес копіювання.
- Процес редагування файлу.
- Процес перегляду файлу.
- Процес виведення інформації про файл/каталог.

Процес зміни параметрів програми взаємодіє з наступними процесами, які відбуваються у розробленій системі:

- Процес зміни стилю відображення (таблицею чи іконками).
- Процес приховання/показу функціональних клавіш.

КБПЗ-2023

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

Наступним кроком є вибір тієї, або іншої операції над файлами, або каталогами:

- Заміна диска.
- Перехід у інший каталог.
- Виділення файлу/папки.
- Переглядання файлу.
- Редагування файлу.
- Копіювання.
- Видалення.

Розглянемо ці операції більш детально.

Якщо необхідно змінити диск, то виконуються наступні дії:

- Змінюється диск.
- Виводиться вміст кореневого каталогу диску.

Якщо необхідно перейти у інший каталог, то виконуються наступні дії:

- Відбувається перехід у вказаний каталог.
- Виводиться вміст вказаного каталогу.

Якщо необхідно виділити файл/папку, то відбувається виконання наступних кроків:

- Виводиться інформація про виділений файл/папку.

Якщо необхідно переглянути файл, то відбувається відкриття та перегляд виділеного файлу.

Якщо необхідно редагувати файл, то відбувається відкриття та редагування файлу.

Якщо необхідно копіювати/перемістити файл або каталог, то відбувається копіювання або переміщення виділених файлів або каталогів.

Якщо необхідно видалити файли або каталоги, то відбувається видалення виділених файлів або каталогів.

Після усіх вище перерахованих операцій відбувається виведення оновленого стану каталогів.

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

Після цього користувач обирає, працювати йому далі з програмним продуктом, або ні.

На рисунку 4.2 зображено блок-схему алгоритму роботи підпрограми виведення початкового стану каталогів.

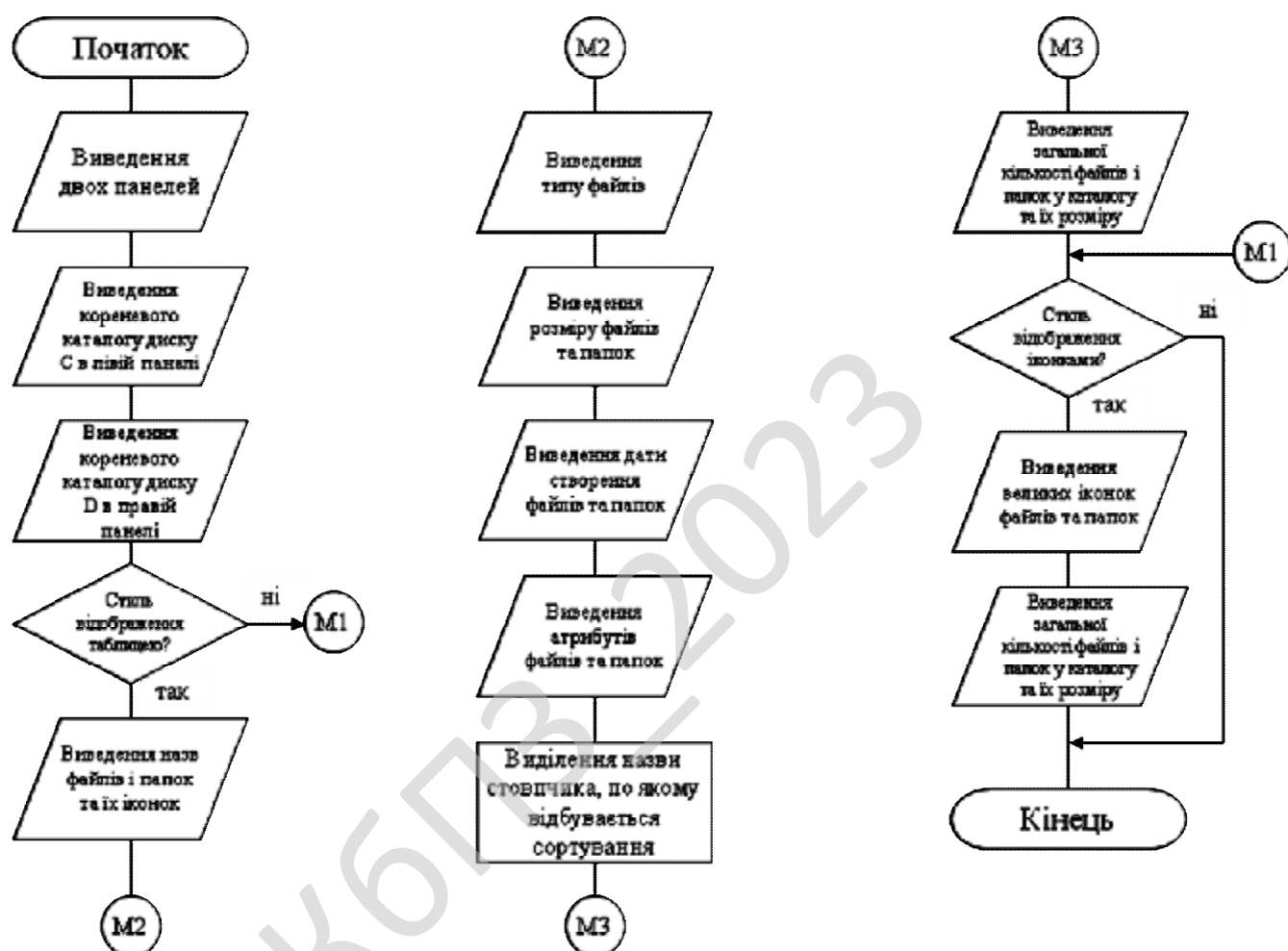


Рисунок 4.2 – Блок-схема алгоритму роботи підпрограми виведення початкового стану каталогів

Ця підпрограма працює наступним чином. Спершу відбувається виведення двох панелей, на яких знаходяться файли й каталоги. Та обрана інформація про них.

При цьому, за замовчування на лівій панелі виводиться кореневий каталог диску C, на правій панелі виводиться кореневий каталог диску D.

Далі обирається вид виведення файлів.

Якщо обрано стиль відображення таблицею, то відбувається виконання наступних операцій:

- Виведення назв файлів і папок та їх іконок.
- Виведення розміру файлів та папок.
- Виведення дати створення файлів та папок.
- Виведення типу файлів.
- Виведення атрибутів файлів та папок.
- Виділення назви стовпчика, по якому повинно відбуватися сортування.
- Виведення загальної кількості файлів і папок у каталозі, та їх розміру.

Якщо обрано стиль відображення таблицею, то відбувається виконання наступних операцій:

- Виведення великих іконок файлів та папок.
- Виведення загальної кількості файлів і папок у каталозі, та їх розмір.

Розглянувши блок-схему основної програми та блок-схему алгоритму роботи підпрограми виведення початкового стану каталогів перейдемо до опису процесу реалізації програмного продукту.

Головна форма програми

При запуску програми створюється головна форма Form1 – представник класу TForm. При створенні форми створюються списки доступних пристроїв, відображувані в FileListLeft і FileListRight, в обох панелях відображається коренева директорія диска C:\, оскільки нами було зроблене припущення, що цей логічний диск повинен бути практично на всіх комп'ютерах, так само створюється безліч атрибутів директорії DirAttribSet, використовуване функціями другого модуля. В Form1 реалізована одна подія: Create (створення, функція-оброблювач: FormCreate). Опишемо об'єкти, розташовані на Form1.

FileListLeft і FileListRight – представники класу TStringGrid (строкова таблиця). Виконують функцію панелей, куди відбувається відображення вмісту директорій і де користувач вибирає файл (каталог), над яким він збирається виконувати яку-небудь дію. Тут велике значення має поняття «активна панель».

Роль активної панелі грає та таблиця, до якої в цей момент звертається користувач. Всі операції відбуваються з файлами (каталогами), відображеними в активній панелі (копіювання й переміщення здійснюється в папку, вміст якої в цей момент відображено в неактивній панелі). У програмі передбачений різний вид для активної й неактивної панелей. В FileListLeft і FileListRight реалізовані наступні події: DbClick (подвійний клік миші, функції-оброблювачі: FileListLeftDbClick і FileListRightDbClick), Enter (вхід, що означає, що панель стає активною, функції-оброблювачі: FileListLeftEnter і FileListRightEnter), KeyDown («гарячі клавіші», функції-оброблювачі: FileListLeftKeyDown і FileListRightKeyDown).

ListDevicesLeft і ListDevicesRight – представники класу TComboBox (випадаючий список вибору). Є списками доступних пристроїв зберігання інформації. Вибір користувачем якого-небудь пристрою із цього списку означає перехід до кореневої директорії цього пристрою, що відображається у відповідній панелі, що стає при цьому активною. У випадку, якщо кореневі директорії в обох панелях після вибору збігаються, то в активній панелі відображається вміст тієї ж папки, що й у неактивній. В ListDevicesLeft і ListDevicesRight реалізовані наступні події: Select (вибір пункту в списку, функції-оброблювачі: ListDevicesLeftSelect і ListDevicesRightSelect), Enter (вхід, що означає, що відповідна панель стає активною, функції-оброблювачі: FileListLeftEnter і FileListRightEnter).

bRemove, bCopy, bCreateDir, bDelete, bCurrentRootDirLeft і bCurrentRootDirRight – представники класу TButton (кнопка). Кнопки виконують різні функції. У всіх кнопках реалізована тільки одна подія – Click (натискання, функції оброблювачі: bCreateDirClick, bDeleteClick, bCopyClick, bRemoveClick, bCurrentRootDirLeftClick і bCurrentRootDirRightClick). При натисканні на bRemove відбувається переміщення файлу (каталогу, з усіма вкладеними каталогами) з папки, відображеної в активній панелі, у папку, відображену в неактивній. Натискання bCopy практично аналогічно натисканню на bRemove.

тільки відбувається копіювання, а не переміщення. Натискання на `bCreateDir` виводить діалог створення каталогу `FormDialogCreateDir`. Натискання на `bDelete` виводить діалог видалення або приміщення в кошик `FormDialogDelete`. При натисканні на `bCurrentRootDirLeft` або `bCurrentRootDirRight` у відповідних панелях відбувається відображення кореневої директорії поточного пристрою зберігання інформації.

`LabelCurrentPathLeft` і `LabelCurrentPathRight` – представники класу `TLabel` (напис). У цих об'єктах відображаються поточні шляхи панелей.

Діалог створення каталогу

При натисканні на кнопку форми `Form1` `bCreateDir` на екран виводиться форма `FormDialogCreateDir` – представник класу `TForm`.

Опишемо об'єкти розташовані на `FormDialogCreateDir`:

– `edNewDirName` – представник класу `TEdit` (поле введення). Поле, куди користувач уводить ім'я нового каталогу. В `edNewDirName` реалізована подія `KeyDown` («гарячі клавіші», функція-оброблювач `edNewDirNameKeyDown`).

– `Label1` – представник класу `TLabel` (напис). Містить напис «Створити каталог:».

– `bOk` і `bCancel` – представники класу `TButton` (кнопка). В `bOk` і `bCancel` реалізована одна подія: `Click` (натискання, функції-оброблювачі: `bOkClick` і `bCancelClick`). При натисканні на `bOk`, якщо уведене ім'я правильне (тобто не містить зарезервованих символів і слів і не збігається з ім'ям каталогу в активній панелі) чи немає інших помилок (наприклад, немає прав доступу), створюється каталог у папці, відображеній в активній панелі, і форма ховається, у протилежному випадку видається повідомлення про помилку й пропонується повторити введення. При натисканні на `bCancel` форма ховається без виконання яких-небудь інших дій.

Діалог створення видалення або переміщення в корзину

При натисканні на кнопку форми `Form1` `bDelete` на екран виводиться форма `FormDialogDelete` – представник класу `TForm`.

Опишемо об'єкти на FormDialogDelete:

– Label1 – представник класу TLabel (напис). Містить напис «Виберіть дію:».

– bDelete, bCancel, bRecycler – представники класу TButton (кнопка). В bDelete, bCancel, bRecycler реалізована одна подія: Click (натискання, функції-оброблювачі: bDeleteClick, bCancelClick і bRecyclerClick). При натисканні на bDelete файл (каталог), що перебуває у фокусі в активній панелі, видаляється. При натисканні на bRecycler, файл (каталог), що перебуває у фокусі в активній панелі, міститься в корзину. При натисканні на bCancel форма ховається без виконання яких-небудь інших дій.

Принцип роботи програми

Принцип роботи програми відбитий у блок-схемі. Прокоментуємо її.

При запуску програми створюється головна форма, потім додаток очікує дії користувача. Кожній передбаченій дії зіставлена якась функція-оброблювач.

При натисканні на кнопки «Видалити» і «Створити» програма виводить на екран відповідні діалогові вікна, де так само є передбачені дії користувача з об'єктами на цих формах.

Після виконання якої-небудь дії, програма вертається в режим очікування користувальницьких дій.

Для завершення роботи з Менеджером файлів користувачеві досить натиснути на кнопку «Закрити» системного меню, що перебуває в правому верхньому куті вікна програми.

У Додатку Б наведені вихідні тексти всіх модулів програми, а так само вихідний текст заголовного файлу другого модуля (тому що він, на відміну від заголовних файлів інших модулів не був згенерований автоматично).

Всі функції-оброблювачі подій у всіх формах викликають функції реалізації операцій із другого модуля програми.

Але для початку опишемо використовувані в програмі глобальні змінні й константи.

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

(DequeFindRight) – це контейнер бібліотеки стандартних шаблонів, що є чергою deque, елементами якої є структури типу WIN32_FIND_DATA. Посилання на структуру передаються в FindFirstFile і FindNextFile.

– StrucFindToFileProp. Перетворить структуру типу WIN32_FIND_DATA в _FileProperty. Функції передається структура типу WIN32_FIND_DATA і посилання на структуру типу _FileProperty. Не повертає значення. Викликається в ViewDirectory. Тип _FileProperty описується на початку модуля й використовується для відображення результатів роботи ScanDir в FileListLeft (FileListRight), тому що деякі поля WIN32_FIND_DATA не адаптовані для виводу.

– ViewDirectory. Відображає вміст папки, шлях до якої передається в Path[] в активній панелі. Функції віддається рядок, що закінчується нулем (char Path[]), що містить шлях, що передається в ScanDir, і змінна (integer Panel), передана в ScanDir, і залежно від значення якої, результати відображаються в FileListLeft (якщо Panel=LEFT) або FileListRight (якщо Panel=RIGHT). Не повертає значення. Викликається в HandlerOfOpen, CreateFolder, CopyOrRemove і Deleting.

– HandlerOfOpen. Залежно від дій користувача, виконує різні операції. Якщо відбувається вибір в ListDevicesLeft (ListDevicesRight), то відображається коренева директорія, обраного пристрою за умови, що кореневі директорії панелей не співпадають, інакше в активній панелі відображається відкрита папка неактивної панелі. Якщо відбувається натискання на bCurrentRootDirLeft (bCurrentRootDirRight), то відображається коренева директорія поточного пристрою. Якщо відбувається подвійний клік (натискання клавіші Enter) по FileListLeft (FileListRight), то аналізується, яка рядок перебуває зараз у фокусі, і залежно від вмісту, виконуються різні операції. Якщо у фокусі перебуває файл, то він відкривається додатком, з яким асоційований або, якщо цей додаток, виробляє його запуск. Якщо у фокусі перебуває папка, то відображається її вміст. Всі дії відбуваються з активною панеллю. Функції передається рядок, шлях до кореневої

директорії пристрою (AnsiString RootDir), активна панель (integer Panel) і ознака зміни кореневої директорії (integer DirSel). Повертає поточний шлях активної директорії. Використовується функція Windows API ShellExecute.

– CreateFolder. Створює директорію в активній панелі. Функції передається вказівка на те, яка зараз панель активна (integer Panel). Повертає значення типу boolean, залежно від результатів виконання операції. Функція створює каталог у шляху панелі, що є активною. Як поточний шлях у кожній панелі використовуються значення відповідних глобальних змінних. Використовуються функції Windows API: CreateDirectory і MessageBox.

– Deleting. Видаляє або поміщає в корзину об'єкт (файл або папку, з усіма її вкладеними папками), що перебуває у фокусі в активній панелі. У функцію передається ознака активної панелі (integer Panel) і операція (integer Operation), яку треба зробити, тобто видалити або помістити в кошик. Не повертає значення. Використовується функція Windows API SHFileOperation.

– CopyOrRemove. Переміщає або копіює об'єкт (файл або папку, з усіма її вкладеними папками), що перебуває у фокусі в активній панелі в папку неактивної панелі. У функцію передається ознака активної панелі (integer Panel) і операція (integer Operation), яку треба зробити, тобто копіювати або перемістити. Не повертає значення. Використовується функція Windows API SHFileOperation.

4.2 Захист розробленого програмного забезпечення

Захист розробленого програмного забезпечення буде відбуватися за допомогою алгоритму Camellia – блоковий шифр на основі мережі Фейстеля. У криптографії, Camellia – це симетричний ключ блоковий шифр із розміром блоку 128 біт і розмірами ключа 128, 192 і 256 біт. Він був розроблений спільно Mitsubishi Electric і NTT з Японії. Шифр був схвалений для використання ISO / IEC, проектом Європейського Союзу NESSIE і Японським CRYPTREC

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

проект. шифр має рівні безпеки й можливості обробки, порівнянні з Advanced Encryption Standard.

Шифр був розроблений, щоб підходити як для програмних, так і для апаратних реалізацій, від недорогих смарт-карти для високошвидкісних мережних систем. Він є частиною криптографічного протоколу Transport Layer Security (TLS), призначеного для забезпечення безпеки зв'язки в комп'ютерній мережі, такий як Інтернет

Camellia – це шифр Фейстеля з 18 раундами (при використанні 128-бітних ключів) або 24 раундами (при використанні 192– або 256-бітних ключів). Кожні шість раундів застосовується шар логічного перетворення: так звана «FL-функція» або її зворотна. Camellia використовує чотири 8×8 -бітних S-блоку із вхідними й вихідними афіними перетвореннями й логічними операціями. Шифр також використовує введення й вивід відбілювання клавiш. Шар дифузiя використовує лінійне перетворення на основі матриці з номером галузей 5.

Аналіз безпеки

Камелія вважається сучасним надійним шифром. Навіть при використанні параметра меншого розміру ключа (128 біт) вважається неможливим зламати його за допомогою атаки грубої сили на ключі за допомогою сучасних технологій. Немає відомих успішних атак, що значно послабляють шифр. Шифр був схвалений для використання ISO / IEC, проектом Європейського Союзу NESSIE і Японським CRYPTREC проект. Японський шифр має рівні безпеки й можливості обробки, порівнянні із шифром AES/Rijndael.

Camellia – це блоковий шифр, який може бути повністю визначені мінімальними системами багатомірних багаточленів:

– Камелія (а також AES) S-блоки можуть бути описані системою 23 квадратних рівнянь в 80 членах.

– Розклад ключів можна описати 1120 рівняннями в 768 змінні з використанням 3328 лінійних і квадратичних членів.

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

– Увесь блоковий шифр можна описати 5104 рівняннями в 2816 змінні з використанням 14 592 лінійних і квадратичних членів.

– Усього потрібно 6224 рівняння з 3584 змінними з використанням 17 920 лінійних і квадратичних членів.

– Кількість вільних членів становить 11 696, що приблизно таке ж число, що й для AES.

Теоретично, такі властивості можуть дозволити зламати Camellia (і AES) за допомогою алгебраїчної атаки, такий як розширена розріджена лінеаризація, у т Майбутнє за умови, що атака стане можливою.

Хоча Camellia запатентована, вона доступна за безоплатною ліцензією. Це дозволило шифру Camellia стати частиною проекту OpenSSL під ліцензією з відкритим вихідним кодом з листопада 2006 року. Це також дозволило йому стати частиною Mozilla Модуль NSS (Служби мережної безпеки).

Підтримка Camellia була додана в остаточний випуск Mozilla Firefox 3 в 2008 році (за замовчуванням відключене починаючи з Firefox 33 в 2014 році в дусі «Пропозиції по зміні стандартних наборів шифрів TLS, пропонованих браузерами», який був виключено з версії 37 в 2015 році). Pale Moon, відгалуження Mozilla / Firefox, продовжує пропонувати Camellia і розширив свою підтримку, включивши в нього набори Galois / Counter mode (GCM) із шифром, але вилучив GCM знову у випуску 27.2.0, пославшись на очевидну відсутність інтересу до них.

Пізніше, в 2008 році, група розробки релізу FreeBSD оголосила, що цей шифр також був включений в FreeBSD 6.4. Крім того, Йошисато Янагисава додав підтримку шифру Camellia у дисковий клас зберігання geli FreeBSD.

У вересні 2009 року GNU Privacy Guard додала підтримку Camellia у версії 1.4.10.

Veracrypt (відгалуження Truecrypt) включав Camellia як один з підтримуваних алгоритмів шифрування.

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

Крім того, різні популярні бібліотеки безпеки, такі як Crypto ++, Gnutls, mbed TLS і Openssl також включають підтримку Camellia.

26 березня 2013 р. було оголошено, що Camellia була знову обрана для включення в новий список рекомендованих шифрів для електронного уряду Японії як єдиний 128-бітний алгоритм блокового шифрування, розроблений у Японії. Це збігається з тим, що список CRYPTREC обновляється вперше за 10 років. Вибір був заснований на високій репутації Camellia у плані простоти придбання, а також характеристик безпеки й продуктивності, порівнянних з такими з Advanced Encryption Standard (AES). Камелія залишається незмінною у своєму повному втіленні. Неможлива диференціальна атака на Camellia з 12 раундами без шарів FL / FL дійсно існує.

Продуктивність

S-блоки, використовувані Camellia, мають структуру, аналогічну S-блоку AES. У результаті можна прискорити реалізацію програмного забезпечення Camellia за допомогою наборів команд ЦП, розроблених для AES, таких як x86 AES-NI.

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено головне вікно програми. З цього рисунка ми бачимо, що програма представляє собою взаємодію п'яти основних блоків:

- Блок панелей для виведення файлів й каталогів, та роботи з ними.
- Блок роботи з програмою.
- Блок швидкого доступу до операцій, у вигляді «гарячих» клавіш.
- Блок обирання дисків.
- Блок командного рядка MS-DOS.

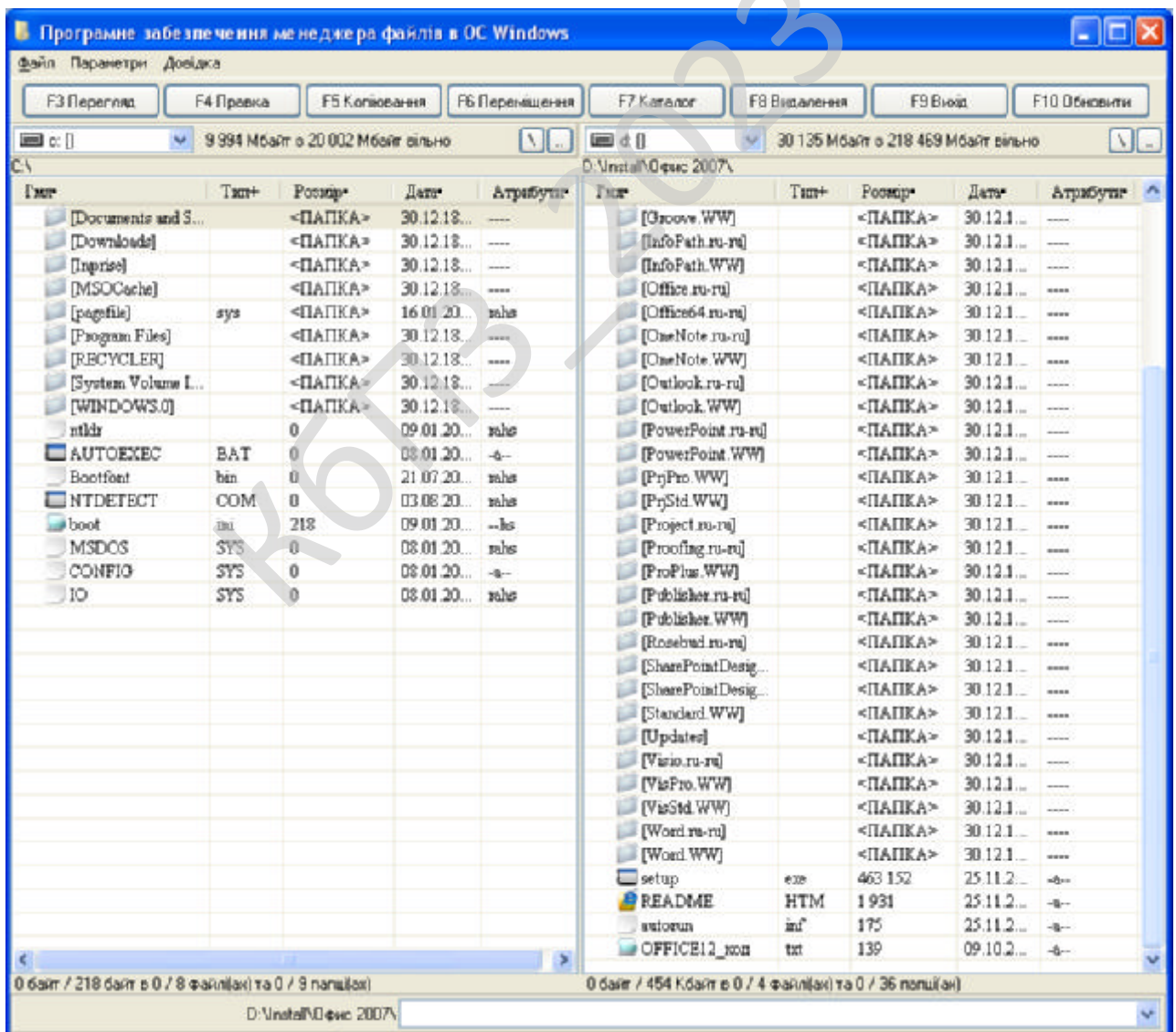


Рисунок 5.1 – Головне вікно програми

На рисунку 5.2 зображено вікно операцій з файлами (копіювання).

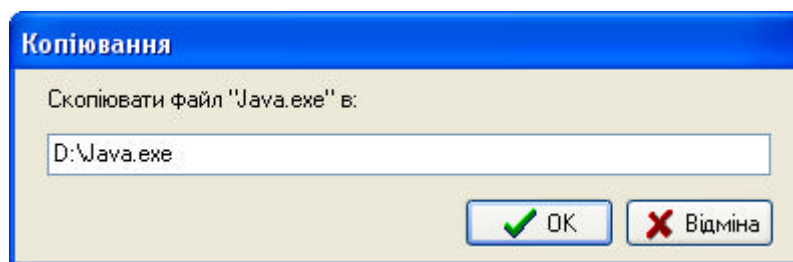


Рисунок 5.2 – Вікно операцій з файлами (копіювання)

На рисунку 5.3 наведено скріншот довідки, у якій наведені «гарячі» клавіші, які дають швидкий доступ до тієї, або іншої операції.

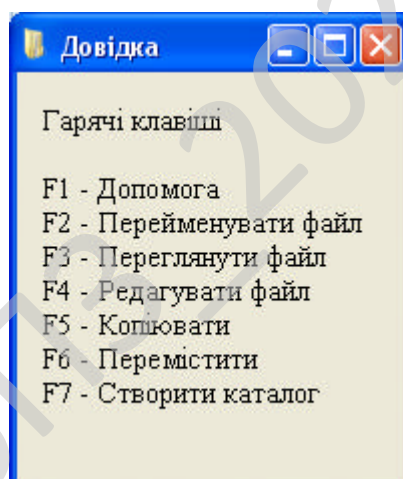


Рисунок 5.3 – Вікно «Довідка»

На рисунку 5.4 наведено вікно інформації про програму, у якому наведені наступні дані:

- Тема проекту.
- Керівник.
- Розробник
- Місце виконання проекту.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Центральноукраїнський національний технічний
університет
Кафедра кібербезпеки та програмного забезпечення
ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему " Дослідження та програмна реалізація системи
менеджменту файлів ОС Windows 11/Windows Server 2022"
ОПП «Комп'ютерні науки»
спеціальності 122 «Комп'ютерні науки»
Виконав: Дудченко А.В.
Науковий керівник: Доренський О.П.
Кропивницький - 2023

ок

Рисунок 5.4 – Вікно «Про програму...»

КБПЗ_2023

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи менеджменту файлів ОС Windows 11/Windows Server 2022.

Метою розробки є дослідження та програмна реалізація системи менеджменту файлів ОС Windows 11/Windows Server 2022.

Об'єктом дослідження є процес менеджменту файлів ОС Windows 11/Windows Server 2022.

Предметом дослідження є методи менеджменту файлів ОС Windows 11/Windows Server 2022.

Методи дослідження базуються на методах файлових систем, методах математичної статистики, методах розробки програмного забезпечення.

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

– Удосконалено метод менеджменту файлів ОС Windows 11/Windows Server 2022.

– Розроблено вітчизняний продукт менеджменту файлів ОС Windows 11/Windows Server 2022, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

Продовження таблиці 7.1

1	2	3
8. Кількість форм вихідної інформації.	–	4
9. Мова програмування (1-6)	–	1
10. Попередній досвід (1-6)	–	3
11. Гнучкість проекту ПП (1-6)	–	3
12. Детальність проекту ПП (1-6)	–	2
13. Рівень спрацьованості колективу (1-6)	–	2
14. Ступінь вимірності процесів (1-6)	–	3
15. Необхідна надійність програмного забезпечення (1-6)	–	2
16. Розмір бази даних (порівняно з розміром програми) (1-6)	–	2
17. Складність кінцевого програмного продукту (1-6)	–	2
18. Необхідний рівень забезпечення повторного використання (1-6)	–	2
19. Документованість відповідно до планованого життєвого циклу (1-6)	–	2
20. Вимоги до швидкодії ПП (1-6)	–	2
21. Обмеження на розміри основного сховища даних (1-6)	–	2
22. Різноманітність використовуваних обчислювальних платформ (1-6)	–	2
23. Професійний рівень аналітиків (1-6)	–	2
24. Професійний рівень програмістів (1-6)	–	2
25. Постійність складу команди розробників (1-6)	–	2
26. Досвід розробки додатків (1-6)	–	2
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 ПО для декількох серверів одночасно (1-6)	–	2
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн	–	60000
33. Норматив додаткової зарплати, % :	Нд	10
34. Норматив відрахувань у соціальні фонди, %	Нс	22
35. Норматив загальногосподарських витрат, %	Нг	15
36. Норматив витрат на освоєння нових мов програмування, %	Нп	15
37. Рівень рентабельності програмної продукції, %	Ре	50
38. Ставка податку на додану вартість, %	Ндв	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 А – коефіцієнт Боема, А=2,45;

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.

$$B = 1,01 + 0,001(2,43 + 3,64 + 3,38 + 3,95 + 2,73) = 1,026$$

$$T_{ном} = 2,45 \cdot 2,7^{1,026} = 6,78 \text{ люд-міс.}$$

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

$$T_{уточн} = T_{ном} \cdot \Pi V_j, \quad (7.3)$$

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

$$T_{уточн} = 6,78 \cdot (0,88 \cdot 0,93 \cdot 0,88 \cdot 0,91 \cdot 0,95 \cdot 1 \cdot 1 \cdot 0,87 \cdot 1,22 \cdot 1,16 \cdot 1,1 \cdot 1,1 \cdot 1,12 \cdot 1,1 \cdot 1,1 \cdot 1,1) = 9,37 \text{ люд-міс.}$$

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:

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

де C – визначений емпірично коефіцієнт, запропонований авторами методики, (МВ, додаток 4); S – коефіцієнт стиснення (або подовження) графіка робіт %, що дозволяє коректувати терміни розробки ПО згідно встановленим вимогам. Вибираємо в межах (25...350)%

$$T_{ПП} = 0,3 \cdot 2,66 \cdot 9,37^{0,33+0,2(1,026-1,01)} \cdot 130 = 265 \text{ люд/день}$$

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

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

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	90	11	990	16,5
Монітор	60	11	660	11
Клавіатура	30	11	330	5,5
Маніпулятор «мишка»	30	11	330	5,5
Принтер матричний	60	0	0	0,0
Принтер лазерний	120	2	240	4
Принтер струминний	60	1	60	1
Сканер	20	1	20	0,33
Концентратор-маршрутизатор	30	2	60	1
Кабельні господарства ЛОМ на 1 м.п.	2,5	200	500	8,33
Копіювальний апарат	140	1	140	2,33
Усього за рік:			З _ч	55,49

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:

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

$$\Phi_{op}^c = \frac{55 \cdot 2}{1,2} = 92 \text{ год}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

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

$$Ч_{ел} = 92 / (48 \cdot 8) = 0,24 \text{ ставки}$$

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 2016, серверу доступу ADSL (OC Linux), налаштування ADSL, VPN PPPoE, Frame Relay, Wi-Fi	0,8	0,2
	Налаштування і конфігурування базової станції безпроводного зв'язку (CMTS)	0,2	
	Розробка та впровадження проектів з організації зв'язку між віддаленими об'єктами, ЛОМ	0,2	
	Забезпечення цілодобової роботи зв'язку клієнтів до мережі Інтернет	0,4	
Всього		1,6	

Продовження таблиці 7.4

Посада	Вид роботи	Час	Кількість 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	0,5	0,2
	Створення графічних і стилістичних елементів сайту	0,5	
	Оформлення банерів і промо-сторінок	0,3	
	Розміщення графіки і контенту на Інтернет сторінках	0,3	
Всього		1,6	
Інженер верстальник	Розробка та верстка макетів рекламної продукції та технічної документації	1	0,2
	Верстка друкованих видань	0,2	
	Додрукова підготовка макетів	0,2	
	Розміщення графіки і контенту на Інтернет сторінках	0,2	
Всього		1,6	

Складемо штатний розклад виконавців у таблицю 7.5.

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

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

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

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

Згідно даних інтернет ресурсу DOM.RIA (<https://dom.ria.com>) ціна одного квадратного метра площі, вік якої не перевищує 30 років, по місту складає 500...1600 у.о./ m^2 . Враховуючи, що курс складає 1 у.о. = 38 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:

$$B_{y\partial} = 8 \cdot 8 \cdot 20000 = 1280000 \text{ грн.}$$

Вартість передавальних пристроїв складає 10% від вартості будівель, і у даному випадку вона складе: 128000 грн.

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

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

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

$$I_{nb} = 8 \cdot 3500 = 28000 \text{ грн}$$

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 20.10.23 – джерело

<http://computorg.ua/ru/price.html>

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Персональний комп'ютер		11771
Системний блок		7771
Процесор	Intel Core i5-7400 (4 ядра по 3.0 – 3.5 GHz) Cache Memory 6 MB	-
Системна плата	Gigabyte GA-B250-D3H, 4x USB 3.0, 4x USB 2.0, 8x Audio Ports, 1x LAN (RJ-45), 1x PS/2, 1x HDMI, 1x DP, 1x DVI, 1x VGA	-
Жорсткий диск	240 GB SSD M.2 Samsung Evo + HDD 500 Gb SAMSUNG Barracuda (3.5", 500GB, 32MB, SATA III)	-
Оперативна пам'ять	16 GB DDR4	-
DVD-привод	DVD±RW ASUS DRW-24B5ST Black Bulk	-
Корпус	LogicPower SFF	-
Кардрідер внутрішній	Transcend TS-RDF8K USB 3.0	-
інше	Клавіатура, мишка	Подарунок
Монітор	Монітор BenQ GL2450HM Black	2600
Принтер лазерний	Canon i-SENSYS LBP6030W	2700
Принтер струминний	Epson Stylus Photo P50 (C11CA45341) + USB cable	5500
Сканер	Epson Perfection V37	2800
Копіювальний апарат	Canon i-SENSYS MF217W with Wi-Fi	5965
Пристрій безперебійного живлення	Powercom BNT-600AP USB	1400

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

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

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 випробовування.	Загальна вартість, грн.
Персональні комп'ютери	8	11771	9416,8	103584,8
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Сканери	1	2800	280	3080
Копіюв. апарат	1	5965	596,5	6561,5
Всього	–	–	–	125216,3

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	1280000	-	-
2. Передавальні пристрої	128000	-	-
Всього по групі	1408000	5	70400

Продовження таблиці 7.8

1	2	3	4
Група 4			
3. Обчислювальна техніка	125216	-	-
Всього по групі	125216	50	62608
Група 5,6			
4. Вимірювальні пристрої	5190	-	-
5. Транспортні засоби	143000	-	
6. Господарський інвентар	28000	-	-
Всього по групі	176190	20	35238
7. Нематеріальні активи	60000	10	6000
Разом	$K_p = 1769406$		$A_p = 174246$

Примітка: вартість автомобіля Dacia Logan LS 2008 взята по даним автобазару «Авто-РІА», джерело https://auto.ria.com/uk/auto_dacia_logan_30226404.html, складає 143000 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.

$$Z_o = 619 \cdot 306 / 60 = 3158 \text{ грн}$$

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

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, %

$$Z_d = 3158 \cdot 10 \cdot 0,01 = 316 \text{ грн}$$

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

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

де H_c – відрахування на соціальні потреби, %

$$C_{oc} = 0,01 \cdot 22(3158 + 316) = 764 \text{ грн}$$

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, %

$$G_{ocn} = 3158 \cdot 15 \cdot 0,01 = 474 \text{ грн}$$

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, картриджей, тонеру, грн., N_e – кількість екземплярів програм, шт.

Згідно прийнятих норм на підприємстві $n_{вум}$ приймаємо 0,4 пачки паперу на період розробки. Тоді, враховуючи, що вартість пачки паперу складає $C_n = 200$ грн., визначаємо вартість паперу за період розробки:

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

$$Z_{M1} = 200 \cdot 0,4 = 80 \text{ грн.}$$

Згідно прийнятих норм по комплектації до вартості запам'ятовуючих пристроїв входить вартість CD/DVD дисків. Їх кількість дорівнює кількості коробочних версій запропонованого продукту (приймаємо 10):

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

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

де: C_{δ} – вартість дисків CD/DVD: CDR box – 23,6 грн./шт., DVD-R box – 49 грн./шт.

$$Z_{M2} = 49 \cdot 10 = 490 \text{ грн.}$$

Згідно норм одноразовій заправці підлягають усі друкуючі пристрої становить:

$$Z_{M3} = \sum C_{з.}, \quad (7.18)$$

де: $C_{з.}$ – 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.

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

$$Z_M = (80 + 490 + 1702) / 60 = 38 \text{ грн.}$$

Визначимо витрати на освоєння нових мов програмування або операційні 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, %

$$O_n = 3158 \cdot 15 \cdot 0,01 = 474 \text{ грн}$$

Визначимо витрати на амортизацію основних фондів з урахування загальної річної суми амортизаційних відрахувань та кількості екземплярів програми ($N_e = 60$ прим.)

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

де A_p – загальна річна сума амортизаційних відрахувань, грн.

$$A_m = 174246 \cdot 2 / (60 \cdot 12) = 484 \text{ грн}$$

Повна собівартість ПЗ визначається як сума витрат за попередніми статтям калькуляції

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

$$C_n = 3158 + 316 + 764 + 474 + 38 + 474 + 484 = 5708 \text{ грн.}$$

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

Витрати на технічне обслуговування:

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

де T_p – кількість годин обслуговування системи за рік, год.,

Z_z – заробітна плата обслуговуючого персоналу, грн/год

Після купівлі нового програмного забезпечення кількість профілактичних годин робіт зменшилася з 320 годин на рік до 120 годин на рік, тому витрати на технічне обслуговування зменшилися з

$$Z_{p \text{ баз}} = 320 \cdot 100 \cdot 1,1 \cdot 1,22 = 42944 \text{ грн.}$$

до

$$Z_{p \text{ нов}} = 120 \cdot 100 \cdot 1,1 \cdot 1,22 = 16104 \text{ грн.}$$

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

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

$$Z_{ел \text{ баз}} = 0,2 \cdot 7200 \cdot 1,9 = 2736 \text{ грн}$$

$$Z_{ел \text{ нов}} = 0,2 \cdot 3600 \cdot 1,9 = 1368 \text{ грн}$$

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

Таблиця 7.12 – Розрахунок амортизаційних відрахувань

Групи основних фондів	Норма амортизації ї %	Балансова вартість, грн., за варіантами		Сума відрахувань, грн за варіантами	
		Базовий	Новий	Базовий	Новий
Програмна продукція	50	–	10274	–	5137
Всього відрахувань	-	–	10274	–	5137

$$T_{cn} = \frac{K_n - K_b}{I_b - I_n} \quad (7.28)$$

$$T_{cn} = \frac{10274}{45680 - 22609} = 0,4 \text{ року}$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1. Кількість екземплярів програми	Прим.	60
2. Повна собівартість розробленої програми	Грн.	5708
3. Ціна розробленої програми	Грн.	8562
4. Плановий прибуток від реалізації розробленої програми	Грн.	2854
5. Рентабельність програмної продукції	%	50
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	1769406
7. Загальний прибуток від реалізації програмної продукції	Грн.	171240
8. Величина економічного ефекту при виготовленні програмної продукції	Грн.	142200
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років	0,4
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	10274
11. Величина економічного ефекту у користувача програмної продукції	Грн.	17934
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	0,4

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.

КБПЗ_2023

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Законом України “Про охорону праці” [3] регламентуються загальні положення державної політики в галузі охорони праці, а конкретизуються ці положення нормативно-правовими актами про охорону праці, зокрема Наказом Міністерства соціальної політики України 14.02.2018 № 207, який зареєстровано в Міністерстві юстиції України 25 квітня 2018 р. за №508/31960 «Про затвердження Вимог щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями» [5], яким затверджено нормативно-правовий акт з охорони праці НПАОП 0.00-7.15-18, «Правила охорони праці під час експлуатації електронно-обчислювальних машин», та «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98.

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

Ці шкідливі фактори можуть привести до професійних захворювань.

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

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

Щоб запропонувати заходи щодо зменшення негативного впливу комп'ютера на організм людини визначемо фактори, які можуть викликати професійне захворювання і впливають на працездатність програміста.

8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

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

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

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

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» [5] та НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин»). Таним чином можна зробити висновок, що санітарно-гігієнічні умови праці на робочому місці програміста відповідають вимогам.

Температура повітря в приміщенні визначається впливом температури зовнішнього повітря і тепловою енергією, яка виділяється всередині приміщення. Джерелами виділення теплоти в даному приміщенні є електроустаткування, освітлювальні прилади, а також люди. У світлий час доби джерелом надлишкового тепла є сонячна радіація. Згідно

Постанови № 42 від 01.12.1999 Головного державного санітарного лікаря України, робота, виконувана в даному приміщенні, відноситься до категорії Ia. В цьому випадку людина витрачає енергії до 120 ккал у годину. Вологість повітря в приміщенні визначається впливом багатьох факторів, серед яких: вологість атмосферного повітря, виділення вологи людьми (при диханні та випарами з поверхні шкіри).

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

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

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		87

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

8.4 Розробка заходів з умов поліпшення охорони праці

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

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

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

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

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		89

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

8.5 Розрахункова частина

Освітлення приміщень, де працюють спеціалісти з ІТ, є важливим фактором для забезпечення комфорту, продуктивності та здоров'я працівників. Недостатнє або надмірне освітлення може призводити до втоми, стресу, головного болю, порушення зору та інших проблем. Для того, щоб уникнути цих проблем, необхідно дотримуватися таких рекомендацій:

– Вибирати джерела світла, які мають природний спектр, не мерехтять і не випромінюють шкідливе випромінювання. Наприклад, LED-лампи або енергозберігаючі лампи.

– Регулювати яскравість і контрастність монітора так, щоб вони не були надто яскравими або тьмяними. Застосовувати режим “Night Shift” або “Dark Mode” для зменшення навантаження на очі.

– Розташовувати монітор так, щоб він не відбивав світло від вікна або лампи. Уникати прямого сонячного світла на екрані або обличчі.

– Забезпечувати рівномірне освітлення робочого місця, уникаючи тіней і блисків. Використовувати локальне освітлення для покращення видимості документів або інших предметів.

– Дотримуватися норм освітленості для роботи з комп'ютерами, які становлять 300-500 лк. Вимірювати освітленість за допомогою спеціальних приладів або мобільних додатків.

– Робити перерви в роботі з комп'ютером кожні 40-60 хвилин і робити вправи для очей. Дивитися на віддалені об'єкти, моргати часто, закривати очі на кілька секунд.

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

Для розрахунку штучного освітлення приміщень, де працюють спеціалісти ІТ, можна використовувати такі методи:

– Метод коефіцієнта використання світлового потоку – цей метод дозволяє врахувати як прямий, так і відбитий світловий потік від стелі, стін і робочої поверхні. Для цього методу потрібно знати нормовану освітленість, площу приміщення, коефіцієнти запасу, нерівномірності і використання світлового потоку.

– Метод питомої потужності – цей метод полягає в тому, що для кожного типу приміщення і роботи визначається питома потужність освітлювальної установки за таблицями. Для цього методу потрібно знати площу приміщення, коефіцієнт запасу і питомий світловий потік ламп.

– Точковий метод – цей метод базується на розрахунку умовної освітленості, яка створюється одним світильником з лампою світловим потоком 1000 лм. Для цього методу потрібно знати геометричні розміри приміщення, висоту підвісу світильників, коефіцієнти запасу і нерівномірності, а також користуватися ізольоксами умовної горизонтальної освітленості

Виконаємо розрахунок освітлення методом коефіцієнта використання світлового потоку – це метод розрахунку штучного освітлення, який дозволяє визначити кількість і потужність світильників, необхідних для забезпечення заданої освітленості на робочих поверхнях.

Цей метод базується на використанні таких понять, як:

– Світловий потік – це кількість світла, яку випромінює джерело світла за одиницю часу. Вимірюється в люменах (лм).

– Коефіцієнт використання світлового потоку – це відношення світлового потоку, який досягає робочої поверхні, до загального світлового потоку, який випромінюють світильники. Залежить від геометрії приміщення, характеристик світильників і коефіцієнтів відбиття поверхонь.

– Освітленість – це кількість світлового потоку, яка припадає на одиницю площі робочої поверхні. Вимірюється в люксах (лк).

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

освітленн. Зазвичай коефіцієнт запасу приймається в межах 1,1-1,5. Приймаємо 1,1.

Для розрахунку кількості світильників в приміщенні, вам потрібно врахувати рівень освітлення, світловий потік кожного світильника, коефіцієнт використання і площу приміщення.

Загальний алгоритм розрахунку:

– **Визначте бажаний рівень освітлення (E):** Це рівень освітлення, який вам потрібно в приміщенні. Він вимірюється в люксах (лк). Для офісних приміщень зазвичай рекомендується рівень освітлення близько 300-500 лк, залежно від конкретних потреб.

– **З'ясуйте світловий потік кожного світильника (F):** Це кількість світла, яке виділяє кожен світильник і вимірюється в люменах (лм). Інформацію про світловий потік зазвичай надає виробник світильника.

– **Визначте коефіцієнт використання (K_c):** Цей коефіцієнт вказує на частину світла, яка досягає поверхні, де працюють співробітники. Виробники світильників також надають інформацію про цей параметр.

– **Визначте виправлення світлового потоку (K):** Виправлення світлового потоку враховує втрати світла через пил, забруднення та відбиття від поверхонь. Для офісних приміщень, де чистота зазвичай добра, цей коефіцієнт може бути близько 0,9.

– **Виміряйте площу приміщення (S):** Виміряйте площу приміщення, в якому планується розташування світильників, в квадратних метрах (м²).

– **Використовуйте формулу для розрахунку кількості світильників (N):** Використовуючи вище наведені значення, використовуйте наступну формулу:

$$N = \frac{E \cdot S \cdot Z}{F \cdot K \cdot K_c} \quad (1).$$

Визначемо кількості світильників:

Результат розрахунку дозволить визначити кількість світильників, яку потрібно встановити в приміщенні, щоб досягти бажаного рівня освітлення.

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

Якщо бажаний рівень освітлення $E = 500$ лк, світловий потік кожного світильника $F = 800$ лм, коефіцієнт використання $K = 0,8$, виправлення світлового потоку $K_c = 0,9$, і площа приміщення $S = 50$ м², то кількість світильників буде:

$$N = \frac{500 \text{ лм} \cdot 50 \text{ м}^2 \cdot 1.1}{800 \cdot 0.8 \cdot 0.9} = 47.7.$$

Отже, вам потрібно встановити приблизно 48 світильників з світлодіодними лампами потужністю 10 Вт, які мають світловий потік приблизно 800 лм світильників, щоб досягти бажаного рівня освітлення у цьому приміщенні.

8.6 Висновки до розділу

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

У розділі здійснено аналіз загальних умов праці, призначеного для праці програмістів, розглянуто та проаналізовано небезпечні та шкідливі фактори, які негативно впливають на ІТ спеціалістів під час професійної діяльності.

Виконано розрахунок штучного освітлення, як одного з ключових факторів впливу на працездатність та здоров'я ІТ спеціаліста. Розроблено заходи з охорони праці.

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		94

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи менеджменту файлів ОС Windows 11/Windows Server 2022.

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 менеджменту файлів ОС Windows 11/Windows Server 2022.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем менеджменту файлів ОС Windows 11/Windows Server 2022.

– Досліджена система менеджменту файлів ОС Windows 11/Windows Server 2022.

– На основі отриманих результатів досліджень створена програмна реалізація системи менеджменту файлів ОС Windows 11/Windows Server 2022.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання менеджменту файлів ОС Windows 11/Windows Server 2022.

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

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

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		95

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

Програма реалізована на мові високого рівня Delphi 10.4. 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 11/Windows Server 2022.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм Camellia.

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 17934 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 0,4 роки.

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		96

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Дудченко А.В. Дослідження та програмна реалізація системи менеджменту файлів ОС Windows 11/Windows Server 2022 // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2023.
2. Wendell Odom. «CCNA 200-301 Official Cert Guide, Volume 1». Cisco Press. 2020. – 848 p.
3. Wendell Odom. «CCNA 200-301 Official Cert Guide, Volume 2 Premium Edition eBook and Practice Test». Cisco Press. 2020. – 624 p.
4. Scott Jernigan «CompTIA Network+ Certification All-in-One Exam Guide, Eighth Edition». 2022. – 976 p.
5. Doug Lowe «Networking For Dummies 12th Edition». 2020. – 480 p.
6. Ramon Nastase «Computer Networking: The Beginner's guide for Mastering Computer Networking, the Internet and the OSI Model». 2018. – 186 p.
7. Russ White & Ethan Banks «Computer Networking Problems and Solutions: An Innovative Approach to Building Resilient, Modern Networks». 2017. – 832 p.
8. 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.
9. 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.
10. Smirnova T., Gnatyuk S., Berdibayev R., Avkurova Zh., Iavich M. «Cloud-Based Cyber Incidents Response System and Software Tools».

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

Communications in Computer and Information Science, 2021, vol 1486. Springer, Cham. pp 169-184.

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

12. 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.

13. 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.

14. 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.

15. 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.

16. 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.

17. 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.

					БКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		98

18. 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.

19. 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.

20. 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.

21. 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.

22. 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.

23. 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.

24. 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.

25. 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.

26. 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.

27. 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.

28. 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.

29. 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.

30. 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.

31. 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.*

32. 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.*

33. Смірнов О.А., Смірнова Т.В., Буравченко К.О., Кравченко С.С., Горбов В.О., «Хмарна система підтримки прийняття рішень технологічного процесу відновлення поверхонь конструкцій і деталей машин». *Сучасні інформаційні системи. 2021. Т. 5, № 4. С. 79-95*

34. 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.*

35. 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.*

36. 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.

37. 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.*

38. 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.*

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		101

39. 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.

40. 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).

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

42. 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.

43. 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.

44. 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.

45. 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.

					ВКРМ-122.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		102

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

47. 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.

48. 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.

49. 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.

50. 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.

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-122.23.0006.00.00.ТЗ		
Вим.	Арк.	№ документа	Підпис	Дата			
Розробив	Дудченко А.В.				Літ.	Аркуш	Аркушів
Перевірів	Доренський О.П.						
Н. Контр.	Коваленко А.С.				ЦНТУ КН-22М-1		
Затв.	Смірнов О.А.						
					Дослідження та програмна реалізація системи менеджменту файлів ОС Windows 11/Windows Server 2022		
					М	1	6

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи менеджменту файлів ОС Windows 11/Windows Server 2022.

2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 32-13 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 менеджменту файлів ОС Windows 11/Windows Server 2022.

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-122.23.0006.00.00.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 менеджменту файлів ОС Windows 11/Windows Server 2022;
- 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-122.23.0006.00.00.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 11/Windows Server 2022 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 11/Windows Server 2022.

5.8.1 Обладнання

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

5.8.2 Мова програмування

Середовище Delphi 10.4.

					ВКРМ-122.23.0006.00.00.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 шкідливі і небезпечні фактори при роботі з комп'ютером.

					ВКРМ-122.23.0006.00.00.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 – 103 аркуші.

10 Етапи розробки

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

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

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

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 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 11.12.2023 р.

					ВКРМ-122.23.0006.00.00.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

_____ Доренський О.П.

*Дослідження та програмна реалізація
системи менеджменту файлів ОС Windows 11/Windows Server 2022*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 36

Літера: РП

Кропивницький – 2023 року

Main.pas - основна програма

```

unit Main;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  FileCtrl, StdCtrls, ComCtrls, frFilePanelU, ImgList, ExtCtrls, Menus,
  XPMan, unCHM;

type
  TFileCommander = class(TForm)
    frFilePanelLeft: TfrFilePanel;
    pnTop: TPanel;
    FileListBox1: TFileListBox;
    ImageList1: TImageList;
    splPanelsSplitter: TSplitter;
    frFilePanelRight: TfrFilePanel;
    mmMenu: TMainMenu;
    miFile: TMenuItem;
    miHelp: TMenuItem;
    miExit: TMenuItem;
    pnBottom: TPanel;
    cmbxCommand: TComboBox;
    lbPath: TLabel;
    lbItem: TLabel;
    miAbout: TMenuItem;
    btEdit: TButton;
    btCopy: TButton;
    btMove: TButton;
    btFolder: TButton;
    btDelete: TButton;
    XPManifest1: TXPManifest;
    Button1: TButton;
    Button3: TButton;
    N1: TMenuItem;
    N2: TMenuItem;
    N3: TMenuItem;
    N4: TMenuItem;
    Button4: TButton;
    N5: TMenuItem;
    ColorDialog1: TColorDialog;
    N6: TMenuItem;
    N7: TMenuItem;
    N8: TMenuItem;
    N9: TMenuItem;
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure FormActivate(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure miExitClick(Sender: TObject);
    procedure cmbxCommandKeyDown(Sender: TObject; var Key: Word;
      Shift: TShiftState);
    procedure cmbxCommandEnter(Sender: TObject);
    procedure miAboutClick(Sender: TObject);
    procedure btEditClick(Sender: TObject);
    procedure btCopyClick(Sender: TObject);
    procedure btMoveClick(Sender: TObject);
    procedure btFolderClick(Sender: TObject);
    procedure btDeleteClick(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure N2Click(Sender: TObject);
    procedure N4Click(Sender: TObject);
    procedure N3Click(Sender: TObject);
    procedure frFilePanelLeftbbRefreshClick(Sender: TObject);
  end;

```

```

    procedure frFilePanelRightbbRefreshClick(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure N5Click(Sender: TObject);
    procedure N6Click(Sender: TObject);
    procedure N9Click(Sender: TObject);
    procedure N8Click(Sender: TObject);
private
    { Private declarations }
    chm:TChm;
public
    { Public declarations }
end;

var
    FileCommander: TFileCommander;

implementation

uses FilesExU, About, Unit1;

Var
    FirstRun:Boolean;

{$R *.DFM}

// Закриття форми

procedure TFileCommander.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    frFilePanelLeft.Done;
    frFilePanelRight.Done;
end;

//Активация форми

procedure TFileCommander.FormActivate(Sender: TObject);
begin
    If FirstRun Then
        Begin
            frFilePanelRight.Init(FileListBox1, ImageList1, frFilePanelLeft.Deactivate,
lbPath, lbItem);
            frFilePanelLeft.Init(FileListBox1, ImageList1, frFilePanelRight.Deactivate,
lbPath, lbItem);
            FirstRun:=False;
        End;
end;

//Сворення форми

Procedure TFileCommander.FormCreate(Sender: TObject);
var
s:string;
begin
    FirstRun:=True;
    s := ExtractFileDir(Application.ExeName);
    if not(s[Length(s)] in ['\','/']) then s:=s+'\';
    s:=s+'help.chm';
    chm := TChm.Create(s);
end;

//Кнопка виходу

procedure TFileCommander.miExitClick(Sender: TObject);
begin
    Close;
end;

```

```

//Командні клавіші

procedure TFileCommander.cmbxCommandKeyDown(Sender: TObject;
  var Key: Word; Shift: TShiftState);
Var
  P:Integer;
  Command, Params:String;
begin
  If Key<>VK_Return Then Exit;
  cmbxCommand.Text:=Trim(cmbxCommand.Text);
  If ssCtrl In Shift Then
  Begin
    cmbxCommand.Text:=' '+cmbxCommand.Text+lbItem.Caption+' ';
    Exit;
  End;

  If cmbxCommand.Items.IndexOf(cmbxCommand.Text)<0 Then
    cmbxCommand.Items.Add(cmbxCommand.Text);
  P:=Pos(' ', cmbxCommand.Text);
  If P<=0 Then
  Begin
    ExecuteOneFile(lbPath.Caption, cmbxCommand.Text, '');
    Exit;
  End;

  Command:=Copy(cmbxCommand.Text, 1, P-1);
  Params:=Copy(cmbxCommand.Text, P+1, Length(cmbxCommand.Text)-P);
  ExecuteOneFile(lbPath.Caption, Command, Params);
end;

//Введення команд

procedure TFileCommander.cmbxCommandEnter(Sender: TObject);
begin
  cmbxCommand.SelectAll;
end;

//Довідка про користувача

procedure TFileCommander.miAboutClick(Sender: TObject);
begin
  PnTop.Visible:=false;
  miAbout.Checked:=true;
  N4.Checked:=false;
end;

// Вхід у режим редагування

procedure TFileCommander.btEditClick(Sender: TObject);
begin
  If frFilePanelLeft.NowActive Then
    frFilePanelLeft.EditFile
  Else
    frFilePanelRight.EditFile;
end;

//Кнопка копіювання

procedure TFileCommander.btCopyClick(Sender: TObject);
begin
  If frFilePanelLeft.NowActive Then
    frFilePanelLeft.TryCopyFile
  Else
    frFilePanelRight.TryCopyFile;
end;

```

```
//Кнопка перенесення файлів або каталогів

procedure TFileCommander.btMoveClick(Sender: TObject);
begin
  If frFilePanelLeft.NowActive Then
    frFilePanelLeft.TryMoveFile
  Else
    frFilePanelRight.TryMoveFile;
end;

//Вхід у каталог

procedure TFileCommander.btFolderClick(Sender: TObject);
begin
  If frFilePanelLeft.NowActive Then
    frFilePanelLeft.CreateFolder
  Else
    frFilePanelRight.CreateFolder;
end;

//Видалення виділених файлів або каталогів

procedure TFileCommander.btDeleteClick(Sender: TObject);
begin
  If frFilePanelLeft.NowActive Then
    frFilePanelLeft.TryDelete
  Else
    frFilePanelRight.TryDelete;
end;

//Процедури обробки кнопок швидкого доступу

procedure TFileCommander.Button1Click(Sender: TObject);
begin
  FileCommander.Close;
end;

procedure TFileCommander.Button3Click(Sender: TObject);
begin
  If frFilePanelLeft.NowActive Then
    frFilePanelLeft.EditFile
  Else
    frFilePanelRight.EditFile;
end;

procedure TFileCommander.Button2Click(Sender: TObject);
begin
  Form1.Show();
end;

procedure TFileCommander.N2Click(Sender: TObject);
begin
  Form1.Show();
end;

procedure TFileCommander.N4Click(Sender: TObject);
begin
  Pntop.Visible:=true;
  miAbout.Checked:=false;
  N4.Checked:=true;
end;

procedure TFileCommander.N3Click(Sender: TObject);
begin
  FileAbout.Show();
end;
```

```

//Виведення лівої панелі файлів

procedure TFileCommander.frFilePanelLeftbbRefreshClick(
  Sender: TObject);
begin
  frFilePanelLeft.bbRefreshClick(Sender);

end;

//виведення правої панелі файлів

procedure TFileCommander.frFilePanelRightbbRefreshClick(
  Sender: TObject);
begin
  frFilePanelRight.bbRefreshClick(Sender);

end;

//процедури обробки натискань

procedure TFileCommander.Button4Click(Sender: TObject);
begin
  frFilePanelRight.bbRefreshClick(Sender);
  frFilePanelLeft.bbRefreshClick(Sender);
end;

procedure TFileCommander.N5Click(Sender: TObject);
begin
  if colordialog1.Execute then
  If frFilePanelLeft.lvFiles.Focused Then
  FileCommander.frFilePanelLeft.lvFiles.Color:=colordialog1.Color
  else
  FileCommander.frFilePanelRight.lvFiles.Color:=colordialog1.Color;
end;

procedure TFileCommander.N6Click(Sender: TObject);
begin
  if colordialog1.Execute then
  If frFilePanelLeft.lvFiles.Focused Then
  FileCommander.frFilePanelLeft.lvFiles.Font.Color:=colordialog1.Color
  else
  FileCommander.frFilePanelRight.lvFiles.Font.Color:=colordialog1.Color;
end;

procedure TFileCommander.N9Click(Sender: TObject);
begin
  If frFilePanelLeft.lvFiles.Focused Then
  FileCommander.frFilePanelLeft.lvFiles.ViewStyle:=vsIcon
  else
  FileCommander.frFilePanelRight.lvFiles.ViewStyle:=vsIcon;
end;

procedure TFileCommander.N8Click(Sender: TObject);
begin
  If frFilePanelLeft.lvFiles.Focused Then
  FileCommander.frFilePanelLeft.lvFiles.ViewStyle:=vsReport
  else
  FileCommander.frFilePanelRight.lvFiles.ViewStyle:=vsReport;
end;

end.

```

frFilePanelU.pas - панель інструментів

```

unit frFilePanelU;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  ComCtrls, StdCtrls, ExtCtrls, FileCtrl,
  FilesU, Buttons, unCHM;

Type
  TColumnsSize=Array [0..4] Of Integer;
  TDeactivateProcedure=Procedure Of Object;
  TfFilePanel = class(TFrame)
    pnDrives: TPanel;
    pnDriveInfo: TPanel;
    lbCurrentPath: TLabel;
    lvFiles: TListView;
    pnFilesInfo: TPanel;
    dcbxDrive: TDriveComboBox;
    btDirRoot: TButton;
    btDirUp: TButton;
    lbDriveInfo: TLabel;
    bbRefresh: TBitBtn;

    procedure lvFilesColumnClick(Sender: TObject; Column: TListColumn);
    procedure dcbxDriveChange(Sender: TObject);
    procedure lvFilesKeyDown(Sender: TObject; var Key: Word;
      Shift: TShiftState);
    procedure lvFilesDblClick(Sender: TObject);
    procedure btDirUpClick(Sender: TObject);
    procedure btDirRootClick(Sender: TObject);
    procedure lvFilesColumnRightClick(Sender: TObject; Column: TListColumn;
      Point: TPoint);
    procedure lvFilesEditing(Sender: TObject; Item: TListItem;
      var AllowEdit: Boolean);
    procedure lvFilesInfoTip(Sender: TObject; Item: TListItem;
      var InfoTip: String);
    procedure lvFilesChange(Sender: TObject; Item: TListItem;
      Change: TItemChange);
    procedure lvFilesEnter(Sender: TObject);
    procedure lvFilesExit(Sender: TObject);
    procedure lbCurrentPathClick(Sender: TObject);
    procedure bbRefreshClick(Sender: TObject);
  private
    { Private declarations }
    AllFiles:TFiles;

    SecondEdit:Boolean;
    Inited:Boolean;

    LastCaption, LastExt:String;
    chm:TChm;
  public
    { Public declarations }
    flbxFiles:TFileListBox;
    CurrentFullPath:String;
    CurrentDrive:Char;
    CurrentPath:String;
    NowRoot:Boolean;

    SortColumn:Byte;
    SortAscending:Boolean;
    NowActive:Boolean;
    UseCopyToDir:String;
  end;
end;

```

```

OtherPanelDeactivate:TDeactivateProcedure;
lbPathEx, lbItemEx:TLabel;

Procedure Init(FilesListBox:TFileListBox; ImageList:TImageList;
Deactivation:TDeactivateProcedure; lbPath, lbItem:TLabel);
Procedure Done;

Procedure MakeOutLabels;
Procedure Activate;
Procedure Deactivate;
Procedure CheckActive;

Procedure Refresh;
Procedure Sort;

Procedure SetPath(Path:String);

Procedure ShowItem(Item:TFileRecord);
Procedure ShowFiles;

Procedure GetItemByList(ListItem:TListItem; Var Item:TFileRecord);

Procedure ShowInfo;
Function ChangeCheck(ListItem:TListItem):Boolean;

Procedure SetColumnsSize(ColumnsSize:TColumnsSize);
Procedure GetColumnsSize(Var ColumnsSize:TColumnsSize);

Procedure SelectLastItem;

Function TryRename(ListItem:TListItem; NewName:String):Boolean;
Function TryOneDelete(Item:TFileRecord):Integer;
Function TryDelete:Boolean;
Procedure TryCopyFile;
Procedure TryMoveFile;

Procedure EditFile;
Procedure CreateFolder;

Procedure SetDrive(Drive:Char);
Procedure CheckCurrentPath;
end;

implementation
Uses
  StrConsts, FilesExU, fmErrorDriveU, fmNameQueryU, fmAnyMessageU, Main,
  About;

{$R *.DFM}

//Ініціалізація файлової панелі

Procedure TfrFilePanel.Init(FilesListBox:TFileListBox; ImageList:TImageList;
Deactivation:TDeactivateProcedure; lbPath, lbItem:TLabel);
Begin
  AllFiles.Init;

  flbxFiles:=FilesListBox;
  lvFiles.LargeImages:=ImageList;
  lvFiles.SmallImages:=ImageList;
  lvFiles.StateImages:=ImageList;
  SortColumn:=1;
  SortAscending:=True;

  SecondEdit:=False;
  Inited:=True;
  LastCaption:='';
  LastExt:='';

```

```

    OtherPanelDeactivate:=Deactivation;
    lbPathEx:=lbPath;
    lbItemEx:=lbItem;
  {}
  Activate;
  SetDrive('C');
End;

Procedure TfrFilePanel.Done;
Begin
  Deactivate;
  AllFiles.Done;
End;

Procedure TfrFilePanel.MakeOutLabels;
Var
  Item:TFileRecord;
Begin
  If lbPathEx<>nil Then
    Begin
      lbPathEx.Caption:=CurrentFullPath;
      lbPathEx.Hint:=CurrentFullPath;
    End;

    If lbItemEx<>nil Then
      Begin
        GetItemByList(lvFiles.ItemFocused, Item);
        lbItemEx.Caption:=GetFullName(Item);
      End;
    End;
End;

Procedure TfrFilePanel.Activate;
Begin
  If @OtherPanelDeactivate<>nil Then OtherPanelDeactivate;
  NowActive:=True;
  UseCopyToDir:=ConstCopyToDir;
  lbCurrentPath.Color:=ConstLabelActiveColor;
  lvFiles.SetFocus;
  MakeOutLabels;
End;

Procedure TfrFilePanel.Deactivate;
Begin
  NowActive:=False;
  ConstCopyToDir:=CurrentFullPath;
  lbCurrentPath.Color:=ConstLabelNonActiveColor;
End;

//Вибір активної файлової панелі

Procedure TfrFilePanel.CheckActive;
Begin

End;

Procedure TfrFilePanel.Refresh;
Var
  i:Integer;
  Item:TFileRecord;
Begin
  CheckCurrentPath;

  AllFiles.Clear;

  flbxFiles.Directory:=CurrentFullPath;
  flbxFiles.Update;
  flbxFiles.FileType:=[ftReadOnly, ftHidden, ftSystem, ftArchive, ftDirectory,
ftNormal];
  If flbxFiles.Items.Count<=0 Then Exit;

```

```

CurrentFullPath:=IncludeTrailingBackslash(flbxFiles.Directory);
CurrentDrive:=ExtractFileDrive(CurrentFullPath)[1];

NowRoot:=IsRoot(CurrentFullPath);

For i:=0 To flbxFiles.Items.Count-1 Do
Begin
  GetItemByFileName(flbxFiles.Items[i], Item);
  If IsDirectory(Item) Then
  Begin
    If ((Item.Name<>'..') And (Item.Name<>'.')) And (Item.Name<>'')) Then
      AllFiles.Add(Item);
    End
  Else Begin
    AllFiles.Add(Item);
  End;
End;
Sort;
ShowFiles;

SelectLastItem;
ShowInfo;

{$I-}
If IOResult<>0 Then MessageBeep(48);
{$I+}
End;

//Сортування файлів на панелі за ознакою

Procedure TfrFilePanel.Sort;
Var
  i:Integer;
  ColCapt:String;
Begin
  For i:=0 To lvFiles.Columns.Count-1 Do
  Begin
    ColCapt:=lvFiles.Column[i].Caption;
    Delete(ColCapt, Length(ColCapt), 1);
    lvFiles.Column[i].Caption:=ColCapt+ConstNoSort;
  End;
  ColCapt:=lvFiles.Column[SortColumn].Caption;
  Delete(ColCapt, Length(ColCapt), 1);
  If SortAscending Then
    lvFiles.Column[SortColumn].Caption:=ColCapt+ConstSortAscending
  Else
    lvFiles.Column[SortColumn].Caption:=ColCapt+ConstSortDescending;

  Case SortColumn Of
    0: AllFiles.SortByName(SortAscending);
    1: AllFiles.SortByExt(SortAscending);
    2: AllFiles.SortBySize(SortAscending);
    3: AllFiles.SortByDateTime(SortAscending);
    4: AllFiles.SortByAttr(SortAscending);
  Else
    AllFiles.SortByName(SortAscending);
  End;
End;

procedure TfrFilePanel.lvFilesColumnClick(Sender: TObject;
  Column: TListColumn);
begin
  If lvFiles.ItemFocused<>nil Then
  Begin
    LastCaption:=lvFiles.ItemFocused.Caption;
    LastExt:=lvFiles.ItemFocused.SubItems[0];
  End
  Else Begin

```

```

    LastCaption:='',
    LastExt:='',
End;

If Column.Index=SortColumn Then
    SortAscending:=Not (SortAscending)
Else
    SortAscending:=True;
SortColumn:=Column.Index;
Sort;
ShowFiles;

SelectLastItem;
end;

//Вибір шляху до файлу

Procedure TfrFilePanel.SetPath(Path:String);
Var
    TmpStr:String;
Begin
    TmpStr:=ExcludeTrailingBackslash (Path);
    TmpStr:=ExtractFileName (TmpStr);

    If TmpStr='..' Then
        Begin
            LastCaption:=ExcludeTrailingBackslash (CurrentFullPath);

LastCaption:=ConstDirLeftBracket+ExtractFileName (LastCaption)+ConstDirRightBrack
et;
            LastExt:='';
            End;

            CurrentFullPath:=Path;

            Refresh;
End;

Procedure TfrFilePanel.ShowItem(Item:TFileRecord);
Var
    FormattedItem:TFileFormattedRecord;
    ListItem:TListItem;
Begin
    GetFormattedItem (Item, FormattedItem);
    ListItem:=lvFiles.Items.Add;
    With ListItem Do
        Begin
            ImageIndex:=GetItemImageIndex (Item);
            If Item.Checked Then
                StateIndex:=0
            Else
                StateIndex:=-1;
            If IsDirectory (Item) Then
                Caption:=ConstDirLeftBracket+FormattedItem.Name+ConstDirRightBracket
            Else
                Caption:=FormattedItem.Name;
            SubItems.Add (FormattedItem.Ext);
            SubItems.Add (FormattedItem.Size);
            SubItems.Add (FormattedItem.DateTime);
            SubItems.Add (FormattedItem.Attr);
        End;
End;

End;

Procedure TfrFilePanel.ShowFiles;
Var
    i:Integer;
    Item:TFileRecord;

```

```

Begin
  lvFiles.Items.Clear;

  If NowRoot Then
    lvFiles.AllocBy:=AllFiles.ItemsCount
  Else
    lvFiles.AllocBy:=AllFiles.ItemsCount+1;

  If Not(NowRoot) Then ShowItem(DirUpItem);

  For i:=0 To AllFiles.ItemsCount-1 Do
  Begin
    AllFiles.GetItem(i, Item);

    ShowItem(Item);

  End;

End;

Procedure TfrFilePanel.GetItemByList(ListItem:TListItem; Var Item:TFileRecord);
Var
  i:Integer;
Begin
  i:=lvFiles.Items.IndexOf(ListItem);
  If Not(NowRoot) Then Dec(i);
  If i<0 Then
    Item:=DirUpItem
  Else
    AllFiles.GetItem(i, Item);
End;

//Виведення інформації про файли та папки

Procedure TfrFilePanel.ShowInfo;
Var
  TotalBytes, TotalFree:Int64;
  TotalDirs, TotalFiles, CheckedDirs, CheckedFiles:Integer;
  TotalSize, CheckedSize:Int64;
  TmpStr:String;
Begin
  GetDiskSize(CurrentDrive, TotalBytes, TotalFree);
  lbCurrentPath.Caption:=CurrentFullPath;
  lbCurrentPath.Hint:=CurrentFullPath;
  TmpStr:=GetCompactSize(TotalFree)+' з '+GetCompactSize(TotalBytes)+' вільно';
  lbDriveInfo.Caption:=TmpStr;
  lbDriveInfo.Hint:=TmpStr;

  AllFiles.GetInfo(TotalDirs, TotalFiles, CheckedDirs, CheckedFiles, TotalSize,
CheckedSize);
  TmpStr:=' '+GetCompactSize(CheckedSize)+' / '+GetCompactSize(TotalSize)+' в '+
  GetFormattedSize(CheckedFiles, 1, True)+' / '+GetFormattedSize(TotalFiles,
1, True)+' файли(ax) та '+
  GetFormattedSize(CheckedDirs, 1, True)+' / '+GetFormattedSize(TotalDirs, 1,
True)+' папки(ax)';
  pnFilesInfo.Caption:=TmpStr;
  pnFilesInfo.Hint:=TmpStr;
End;

Procedure TfrFilePanel.SetColumnsSize(ColumnsSize:TColumnsSize);
Var
  i:Integer;
Begin
  For i:=0 To lvFiles.Columns.Count-1 Do
    lvFiles.Columns.Items[i].Width:=ColumnsSize[i];
End;

Procedure TfrFilePanel.GetColumnsSize(Var ColumnsSize:TColumnsSize);
Var

```

```

    i:Integer;
Begin
    For i:=0 To lvFiles.Columns.Count-1 Do
        ColumnsSize[i]:=lvFiles.Columns.Items[i].Width;
    End;

    //Зміна диску

procedure TfrFilePanel.dcbxDriveChange(Sender: TObject);
begin
    If Not(Inited) Then Exit;
    SetDrive(dcbxDrive.Drive);
    Refresh;
    Activate;
end;

    //Робота з файлом

procedure TfrFilePanel.lvFilesKeyDown(Sender: TObject; var Key: Word;
    Shift: TShiftState);
Var
    TmpStr:String;
    TmpBool:Boolean;
    s:string;
begin
    Case Key Of
        VK_Space:Begin
            ChangeCheck(lvFiles.ItemFocused);
            End;
        VK_Insert:Begin
            ChangeCheck(lvFiles.ItemFocused);
            End;
        VK_Return:Begin
            lvFilesDblClick(Self);
            End;
        VK_Left:Begin
            lvFiles.ItemFocused:=lvFiles.Items.Item[0];
            lvFiles.Selected:=lvFiles.Items.Item[0];
            End;
        VK_Right:Begin
            lvFiles.ItemFocused:=lvFiles.Items.Item[lvFiles.Items.Count-1];
            lvFiles.Selected:=lvFiles.Items.Item[lvFiles.Items.Count-1];
            End;
        VK_Back:Begin
            If ssCtrl In Shift Then
                btDirRootClick(Self)
            Else
                btDirUpClick(Self);
            End;
        VK_Delete:Begin
            TryDelete;
            End;
        VK_F8:Begin
            TryDelete;
            End;
        VK_F1:Begin
            //FileAbout.show();
            s := ExtractFileDir(Application.ExeName);
            if not(s[Length(s)] in ['\','/']) then s:=s+'\';
            s:=s+'help.chm';
            chm := TChm.Create(s);
            chm.ShowHelp();
            End;
        VK_F2:Begin
            lvFilesEditing(Self, lvFiles.ItemFocused, TmpBool);
            End;
        VK_F3:Begin
            EditFile;
            End;
    End;
end;

```

```

VK_F4:Begin
  If ssShift In Shift Then
    Begin
      TmpStr:=ConstLastRequest;
      If Not (GetNameQuery('Редагувати файл:', TmpStr)) Then Exit;
      ExecuteOneFile (CurrentFullPath, ConstNotepadFile, TmpStr);
      Exit;
    End;
  EditFile;
End;
VK_F5:Begin
  TryCopyFile;
End;
VK_F6:Begin
  TryMoveFile;
End;
VK_F7:Begin
  CreateFolder;
End;
VK_F9:Begin
  FileCommander.close();
End;
VK_F10:Begin
  FileCommander.frFilePanelRight.bbRefreshClick(Sender);
  FileCommander.frFilePanelLeft.bbRefreshClick(Sender);
End;
VK_ADD:Begin
  TmpStr:='*.*';
  If Not (GetNameQuery('Виділити файли:', TmpStr)) Then Exit;

  End;
VK_SUBTRACT:Begin
  TmpStr:='*.*';
  If Not (GetNameQuery('Зняти виділення:', TmpStr)) Then Exit;

  End;
End;

{VK_ADD      Add key (numeric keypad)
VK_SEPARATOR Separator key (numeric keypad)
VK_SUBTRACT Subtract key (numeric keypad)
VK_DECIMAL  Decimal key (numeric keypad)
}
end;

Function TfrFilePanel.ChangeCheck(ListItem:TListItem):Boolean;
Var
  Item:TFileRecord;
  ID:Integer;
Begin
  Result:=False;
  GetItemByList(ListItem, Item);
  If Item.Name=ConstDirUp Then Exit;
  ID:=Item.ID;

  AllFiles.Items[ID].Checked:=Not (AllFiles.Items[ID].Checked);
  If AllFiles.Items[ID].Checked Then ListItem.StateIndex:=ConstCheckedImageIndex
Else ListItem.StateIndex:=ConstUnCheckedImageIndex;
  Result:=AllFiles.Items[ID].Checked;
  ShowInfo;
End;

Procedure TfrFilePanel.SelectLastItem;
Var
  ListItem:TListItem;
  StartIndex:Integer;
Begin
  StartIndex:=0;
  Repeat

```

```

ListItem:=lvFiles.FindCaption(StartIndex, LastCaption, False, False, False);
If ListItem=nil Then
Begin
  lvFiles.ItemFocused:=lvFiles.Items.Item[0];
  lvFiles.Selected:=lvFiles.Items.Item[0];
  LastCaption:='';
  LastExt:='';
  Exit;
End;
If ListItem.SubItems[0]=LastExt Then
Begin
  lvFiles.ItemFocused:=ListItem;
  lvFiles.Selected:=ListItem;
  LastCaption:='';
  LastExt:='';
  Exit;
End;
StartIndex:=ListItem.Index;
Until False;
End;

Function TfrFilePanel.TryRename(ListItem:TListItem; NewName:String):Boolean;
Var
  Item:TFileRecord;
  OldName, Name, Ext:String;
  Tmp:Integer;
  TmpStr:String;
Begin
  Result:=False;
  Name:=ExtractFileName(NewName);
  Ext:=ExtractFileExt(NewName);
  // NewName:=CurrentFullPath+Name;
  If Ext<>' ' Then
  Begin
    Delete(Name, Length(Name)-Length(Ext)+1, Length(Ext));
    Delete(Ext, 1, 1);
  End;
  If Name='' Then Exit;

  GetItemByList(ListItem, Item);
  OldName:=CurrentFullPath+GetFullName(Item);
  Tmp:=RenameOneFile(OldName, NewName);
  If Tmp<0 Then
  Begin
    TmpStr:=GetFileError(Tmp)+#0;
    Application.MessageBox(@TmpStr[1], 'Помилка!', MB_ICONERROR Or MB_OK);
    Exit;
  End;

  If IsDirectory(Item) Then
    LastCaption:=ConstDirLeftBracket+Name+ConstDirRightBracket
  Else
    LastCaption:=Name;
  LastExt:=Ext;
  Result:=True;
End;

Function TfrFilePanel.TryOneDelete(Item:TFileRecord):Integer;
Begin
  If Item.Name=ConstDirUp Then
  Begin
    Result:=F_ER_ERROR;
    Exit;
  End;
  If IsDirectory(Item) Then
    Result:=DeleteOneDir(CurrentFullPath+Item.Name)
  Else
    Result:=DeleteOneFile(CurrentFullPath+GetFullName(Item));
End;

```

```
//Вікно видалення файлу
```

```
Function TfrFilePanel.TryDelete:Boolean;
Var
  ListItem:TListItem;
  Item:TFileRecord;
  Tmp:Integer;
  TmpStr:String;
Begin
  Result:=False;
  ListItem:=lvFiles.ItemFocused;
  GetItemByList(ListItem, Item);
  If Item.Name=ConstDirUp Then Exit;

  TmpStr:='Ви дійсно хочете видалити "'+GetFullName(Item)+'"'+#0;
  If Application.MessageBox(@TmpStr[1], 'Попередження', MB_ICONQUESTION Or
  MB_YESNO)=IDNO Then Exit;

  Tmp:=TryOneDelete(Item);
  If Tmp=F_ER_SUCCESS Then
  Begin
    Result:=True;
    Refresh;
    Exit;
  End;
  TmpStr:=GetFileError(Tmp)+#0;
  Application.MessageBox(@TmpStr[1], 'Помилка!', MB_ICONERROR Or MB_OK);
  Result:=False;
End;
```

```
//Вікно копіювання файлу
```

```
Procedure TfrFilePanel.TryCopyFile;
Var
  Item:TFileRecord;
  FileName, TmpStr:String;
  Tmp:Integer;
Begin
  GetItemByList(lvFiles.ItemFocused, Item);
  If IsDirectory(Item) Then
  Begin
    Application.MessageBox('Неможливо скопіювати папку', 'Помилка!',
  MB_ICONERROR Or MB_OK);
    Exit;
  End;
  FileName:=UseCopyToDir+GetFullName(Item);
  FileNameQuery.Caption:='Копіювання';
  If Not(GetNameQuery('Скопіювати файл "'+GetFullName(Item)+'" в:', FileName))
  Then Exit;
  If ExtractFileName(FileName)=FileName Then
    FileName:=CurrentFullPath+FileName;
  ShowAnyMessage('Копіювання...', 'Копіюється файл
  '"+CurrentFullPath+GetFullName(Item)+'" в "'+FileName+'");
  Tmp:=CopyOneFile(CurrentFullPath+GetFullName(Item), FileName, True);
  HideAnyMessage;
  If Tmp=F_ER_SUCCESS Then
  Begin
    MessageBeep(0);
    LastCaption:=Item.Name;
    LastExt:=Item.Ext;
    Refresh;
    Exit;
  End;
  TmpStr:=GetFileError(Tmp)+#0;
  Application.MessageBox(@TmpStr[1], 'Помилка!', MB_ICONERROR Or MB_OK);
End;
```

```

//Вікно переміщення файлу

Procedure TfrFilePanel.TryMoveFile;
Var
  Item:TFileRecord;
  NewName:String;
Begin
  GetItemByList(lvFiles.ItemFocused, Item);
  If Item.Name=DirUpItem.Name Then Exit;
  NewName:=UseCopyToDir+GetFullName(Item);
  FileNameQuery.Caption:='Переміщення';
  If GetNameQuery('Перемістити "'+GetFullName(Item)+'" в:', NewName) Then
  Begin
    If TryRename(lvFiles.ItemFocused, NewName) Then Refresh;
  End;
End;

//Вікно редагування файлу

Procedure TfrFilePanel.EditFile;
Var
  ListItem:TListItem;
  Item:TFileRecord;
Begin
  ListItem:=lvFiles.ItemFocused;
  GetItemByList(ListItem, Item);
  If IsDirectory(Item) Then Exit;

  ExecuteOneFile(CurrentFullPath, ConstNotepadFile, GetFullName(Item));
End;

//Вікно створення файлу

Procedure TfrFilePanel.CreateFolder;
Var
  FolderName:String;
  Tmp:Integer;
  TmpStr:String;
Begin
  FolderName:=ConstLastRequest;
  FileNameQuery.Caption:='Створення папки';
  If Not(GetNameQuery('Створити папку:', FolderName)) Then Exit;
  If FolderName='' Then Exit;
  Tmp:=CreateOneFolder(CurrentFullPath+FolderName);
  If Tmp=F_ER_SUCCESS Then
  Begin
    LastCaption:=ConstDirLeftBracket+FolderName+ConstDirRightBracket;
    LastExt:='';
    Refresh;
    Exit;
  End;
  TmpStr:=GetFileError(Tmp)+#0;
  Application.MessageBox(@TmpStr[1], 'Помилка!', MB_ICONERROR Or MB_OK);
End;

//Вікно вибору диску

Procedure TfrFilePanel.SetDrive(Drive:Char);
Var
  Dir:String;
Begin
  Repeat
    Drive:=UpCase(Drive);
  {$I-}
  GetDir(Ord(Drive)-64, Dir);
  If Drive=Dir[1] Then
    ChDir(Dir)
  Else Begin
    Dir:=Drive+':\';
  End;
End;

```

```

        ChDir (Dir);
    End;
    If IOResult<>0 Then
    Begin
        Dir:=Drive+':\';
        ChDir (Dir);
    End;
    {$I+}
    If IOResult=0 Then
    Begin
        dcbxDrive.Drive:=Drive;
        CurrentDrive:=Drive;
        CurrentFullPath:=Dir;
        Exit;
    End;
    ChooseNewDrive (Drive);
    Until False;
End;

Procedure TfrFilePanel.CheckCurrentPath;
Var
    Dir:String;
Begin
    {$I-}
    ChDir (CurrentFullPath);
    {$I+}
    If IOResult<>0 Then
    Begin
        SetDrive (ExtractFileDrive (CurrentFullPath) [1]);
    End;
    Dir:=CurrentFullPath+#0;
    SetCurrentDirectory (@Dir [1]);
End;

procedure TfrFilePanel.lvFilesDblClick(Sender: TObject);
Var
    ListItem:TListItem;
    Item:TFileRecord;
    ErrorCode:Integer;
    ErrorString:String;
begin
    ListItem:=lvFiles.ItemFocused;
    If ListItem=nil Then Exit;

    GetItemByList (ListItem, Item);

    If IsDirectory (Item) Then
    Begin
        SetPath (CurrentFullPath+Item.Name);
        Exit;
    End;

    ErrorCode:=ExecuteOneFile (CurrentFullPath, GetFullName (Item), '');
    If ErrorCode>32 Then Exit;
    ErrorString:=GetExecuteError (ErrorCode) + #0;
    Application.MessageBox (@ErrorString [1], 'Помилка!', MB_ICONERROR Or MB_OK);
end;

procedure TfrFilePanel.btDirUpClick (Sender: TObject);
begin
    Activate;
    SetPath (CurrentFullPath+'..');
end;

//Вікно виведення дерева каталогів

procedure TfrFilePanel.btDirRootClick (Sender: TObject);
begin

```

```

    Activate;
    SetPath(IncludeTrailingBackslash(ExtractFileDrive(CurrentFullPath)));
end;

procedure TfrFilePanel.lvFilesColumnRightClick(Sender: TObject;
  Column: TListColumn; Point: TPoint);
begin
  Activate;
  Column.Width:=-1;
end;

procedure TfrFilePanel.lvFilesEditing(Sender: TObject; Item: TListItem;
  var AllowEdit: Boolean);
Var
  FileItem:TFileRecord;
  NewName:String;
begin
  AllowEdit:=False;
  GetItemByList(Item, FileItem);
  If FileItem.Name=DirUpItem.Name Then Exit;
  // If IsDirectory(FileItem) Then Exit;

  NewName:=GetFullName(FileItem);
  If GetNameQuery('Перейменувати "'+NewName+'" в:', NewName) Then
  Begin
    If TryRename(Item, NewName) Then Refresh;
  End;

  AllowEdit:=False;
end;

procedure TfrFilePanel.lvFilesInfoTip(Sender: TObject; Item: TListItem;
  var InfoTip: String);
begin
  { If ((lvFiles.StringWidth(InfoTip)+40<lvFiles.Columns[0].Width) Or
  (lvFiles.Columns[0].Width<0)) Then
  Begin
    InfoTip:='';
    Exit;
  End;
  If Item.SubItems[0]<>' ' Then
    InfoTip:=InfoTip+'.'+Item.SubItems[0];}
end;

procedure TfrFilePanel.lvFilesChange(Sender: TObject; Item: TListItem;
  Change: TItemChange);
{Var
  FileItem:TFileRecord;
  ID:Integer;}
Begin
  MakeOutLabels;
  { GetItemByList(Item, FileItem);
  ID:=FileItem.ID;

  AllFiles.Items[ID].Checked:=Item.Checked;

  If Item.Checked Then
    pnFilesInfo.Caption:=Item.Caption+' Перевірено'
  Else
    pnFilesInfo.Caption:=Item.Caption+' Не перевірено';}
end;

procedure TfrFilePanel.lvFilesEnter(Sender: TObject);
begin
  Activate;
end;

procedure TfrFilePanel.lvFilesExit(Sender: TObject);
begin

```

```
// Деактивація;  
end;  
  
procedure TfrFilePanel.lbCurrentPathClick(Sender: TObject);  
begin  
    Activate;  
end;  
  
procedure TfrFilePanel.bbRefreshClick(Sender: TObject);  
begin  
    Refresh;  
    Activate;  
end;  
  
end.
```

КБПЗ_2023

FilesExU.pas - функції роботи з файлами та каталогами

```

unit FilesExU;

interface
Uses
  SysUtils, FileCtrl, ShellApi, Windows;

Const
  F_ER_SUCCESS=0;
  F_ER_HIMSELF=-1;
  F_ER_EXISTS =-2;
  F_ER_NOT_EXISTS=-3;
  F_ER_DIREXISTS=-4;
  F_ER_NOTCOPY=-128;
  F_ER_ERROR=-255;

  ConstCopyToDir:String='';

Procedure GetDiskSize(CurrentDrive:Char; Var TotalBytes, TotalFree:Int64);
Procedure GetRealDiskSize(Drive:Char; Var TotalBytes, TotalFree:Double);

Function ExecuteOneFile(WorkDir, FileName, Params:String):Integer;
Function GetExecuteError(ErrorCode:Integer):String;

Function CopyOneFile(FromFile, ToFile:String; PrevCheck:Boolean):Integer;
Function GetFileError(ErrorCode:Integer):String;

Function RenameOneFile(OldName, NewName:String):Integer;
Function DeleteOneFile(FileName:String):Integer;
Function DeleteOneDir(FileName:String):Integer;

Function CreateOneFolder(FolderName:String):Integer;

implementation

function GetDiskFreeSpaceEx(lpDirectoryName: PAnsiChar;
  var lpFreeBytesAvailableToCaller : Integer;
  var lpTotalNumberOfBytes: Integer;
  var lpTotalNumberOfFreeBytes: Integer) : boolean;
  stdcall;
  external 'kernel32'
  name 'GetDiskFreeSpaceExA';

//Визначення розміру диску

Procedure GetDiskSize(CurrentDrive:Char; Var TotalBytes, TotalFree:Int64);
Var
  Drive:Byte;
Begin
  CurrentDrive:=UpCase(CurrentDrive);
  Drive:=Ord(CurrentDrive)-64;
  TotalBytes:=DiskSize(Drive);
  TotalFree:=DiskFree(Drive);
End;

Procedure GetRealDiskSize(Drive:Char; Var TotalBytes, TotalFree:Double);
Var
  AvailToCall : integer;
  TheSize : integer;
  FreeAvail : integer;
  TheDrive:String;
Begin
  TheDrive:=Drive+'\'+#0;

```

```

    GetDiskFreeSpaceEx(@TheDrive[1], AvailToCall, TheSize, FreeAvail);
{$IFOPT Q+}
{$DEFINE TURNOVERFLOWON}
{$Q-}
{$ENDIF}
    If TheSize >= 0 then
        TotalBytes := TheSize
    Else
        if TheSize = -1 then
            begin
                TotalBytes := $7FFFFFFF;
                TotalBytes := TotalBytes * 2;
                TotalBytes := TotalBytes + 1;
            end
        else begin
            TotalBytes := $7FFFFFFF;
            TotalBytes := TotalBytes + abs($7FFFFFFF - TheSize);
        end;

    If AvailToCall >= 0 then
        TotalFree := AvailToCall
    else
        if AvailToCall = -1 then
            begin
                TotalFree := $7FFFFFFF;
                TotalFree := TotalFree * 2;
                TotalFree := TotalFree + 1;
            end
        else begin
            TotalFree := $7FFFFFFF;
            TotalFree := TotalFree + abs($7FFFFFFF - AvailToCall);
        end;
End;

Function ExecuteOneFile(WorkDir, FileName, Params:String):Integer;
Begin
    FileName:=FileName+#0;
    WorkDir:=WorkDir+#0;
    Params:=Params+#0;

    Result:=ShellExecute(0, 'open', @FileName[1], @Params[1], @WorkDir[1],
SW_SHOWNORMAL);
End;

//Функція обробки помилок

Function GetExecuteError(ErrorCode:Integer):String;
Begin
    Result:='';
    If ErrorCode>32 Then Exit;
    Case ErrorCode Of
        0 : Result:='Системі не вистачає пам'яті або ресурсів';
        ERROR_FILE_NOT_FOUND : Result:='Файл не знайдений';
        ERROR_PATH_NOT_FOUND : Result:='Шлях не знайдений';
        ERROR_BAD_FORMAT : Result:='Помилка у форматі файлу';
        SE_ERR_ACCESSDENIED : Result:='Доступ до файлу закритий';
        SE_ERR_ASSOCINCOMPLETE: Result:='Файлова асоціація невірна';
        SE_ERR_DLLNOTFOUND : Result:='Динамічна бібліотека не знайдена';
        SE_ERR_NOASSOC : Result:='Відсутнє додаток, пов'язане з даним типом
файлу';
        SE_ERR_OOM : Result:='Недостатньо пам'яті для завершення
операції';
        SE_ERR_SHARE : Result:='Помилка спільного доступу';
    Else
        Result:='Помилка при запуску програми';
    End;
End;

End;

```

```

Function CopyOneFile(FromFile, ToFile:String; PrevCheck:Boolean):Integer;
Begin
  If PrevCheck Then
    Begin
      If FromFile=ToFile Then
        Begin
          Result:=F_ER_HIMSELF;
          Exit;
        End;
      If FileExists(ToFile) Then
        Begin
          Result:=F_ER_EXISTS;
          Exit;
        End;
      End;

      FromFile:=FromFile+#0;
      ToFile:=ToFile+#0;

      If Not(CopyFile(@FromFile[1], @ToFile[1], False)) Then
        Result:=F_ER_NOTCOPY
      Else
        Result:=F_ER_SUCCESS;
    End;

    //Функція обробки помилок

Function GetFileError(ErrorCode:Integer):String;
Begin
  Result:='';
  Case ErrorCode Of
    F_ER_SUCCESS: Result:='';
    F_ER_HIMSELF: Result:='Не можна копіювати файл у себе';
    F_ER_EXISTS : Result:='Такий файл уже існує';
    F_ER_DIREXISTS : Result:='Така папка вже існує';
    F_ER_NOT_EXISTS : Result:='Такий файл відсутній';
    F_ER_NOTCOPY: Result:='Помилка при копіюванні';
    F_ER_ERROR: Result:='Помилка при роботі з файлом';
  End;
End;

Function RenameOneFile(OldName, NewName:String):Integer;
Begin
  If (FileExists(NewName) Or DirectoryExists(NewName)) Then
    Begin
      Result:=F_ER_EXISTS;
      Exit;
    End;
  If MoveFile(PChar(OldName+#0), PChar(NewName+#0)) Then
    Result:=F_ER_SUCCESS
  Else
    Result:=F_ER_NOTCOPY;
  End;
End;

Function DeleteOneFile(FileName:String):Integer;
Begin
  If Not(FileExists(FileName)) Then
    Begin
      Result:=F_ER_NOT_EXISTS;
      Exit;
    End;
  {$I-}
  FileSetAttr(FileName, faArchive);
  IOResult;
  {$I+}
  If SysUtils.DeleteFile(FileName) Then

```

```
        Result:=F_ER_SUCCESS
    Else
        Result:=F_ER_ERROR;
    End;

//Видалення директорії

Function DeleteOneDir(FileName:String):Integer;
Begin
    If Not(DirectoryExists(FileName)) Then
        Begin
            Result:=F_ER_NOT_EXISTS;
            Exit;
        End;
    If RemoveDir(FileName) Then
        Result:=F_ER_SUCCESS
    Else
        Result:=F_ER_ERROR;
    End;

//Створення директорії

Function CreateOneFolder(FolderName:String):Integer;
Begin
    If DirectoryExists(FolderName) Then
        Begin
            Result:=F_ER_DIREXISTS;
            Exit;
        End;
    If CreateDir(FolderName) Then
        Result:=F_ER_SUCCESS
    Else
        Result:=F_ER_ERROR;
    End;

end.
```

КБПЗ_2023

FilesU.pas – робота з файлами з різними розширеннями

```

unit FilesU;

interface

Uses
  classes, SysUtils, FileCtrl,
  StrConsts;

Const
  MaxFilesCount=10240;
  MaxFileTypes=41;

Type
  TImagesIndex=Array [0..MaxFileTypes-1, 0..1] Of String;

  TRecordID=Object
    ID:Integer;
  End;
  TFileRecord=Record
    ID:Integer;
    Name:String;
    Ext:String;
    Size:Integer;
    DateTime:TDateTime;
    Attr:Integer;
    Checked:Boolean;
    Tag:Integer;
  End;
  TFileFormattedRecord=Record
    Name,
    Ext,
    Size,
    DateTime,
    Attr:String;
    Checked:Boolean;
    Tag:Integer;
  End;

  TFiles=Object
  Protected
    s1Dirs, s1Files:TStringList;
    DirsID, FilesID:Array[0..MaxFilesCount-1] Of TRecordID;

    Procedure FinishSort(Ascending, DirAscending:Boolean);
  Public
    ItemsID:Array[0..MaxFilesCount-1] Of Integer;
    Items:Array[0..MaxFilesCount-1] Of TFileRecord;
    ItemsCount:Integer;
    Tag:Integer;

    Constructor Init;
    Destructor Done;

    Function Add(Item:TFileRecord):Boolean;
    Procedure Clear;
    Procedure GetItem(ItemNo:Integer; Var Item:TFileRecord);

    Procedure SortByName(Ascending:Boolean);
    Procedure SortByExt(Ascending:Boolean);
    Procedure SortBySize(Ascending:Boolean);
    Procedure SortByDateTime(Ascending:Boolean);
    Procedure SortByAttr(Ascending:Boolean);
    Procedure GetInfo(Var TotalDirs, TotalFiles, CheckedDirs,
CheckedFiles:Integer; Var TotalSize, CheckedSize:Int64);

```

```

End;

Const
  DirUpItem:TFileRecord= (
    ID:0;
    Name:ConstDirUp;
    Ext:'';
    Size:0;
    DateTime:0;
    Attr:faDirectory;
    Checked:False;
    Tag:0);

  Images:TImagesIndex= (
    ('', '19'),
    ('EXE', '14'),
    ('BAT', '14'),
    ('COM', '14'),
    ('LNK', '19'),
    ('PIF', '17'),
    ('TXT', '20'),
    ('LOG', '20'),
    ('HTM', '21'),
    ('HTML', '21'),
    ('MHT', '21'),
    ('DOC', '22'),
    ('DOT', '22'),
    ('XLS', '23'),
    ('RAR', '26'),
    ('ZIP', '27'),
    ('ARJ', '27'),
    ('PAS', '28'),
    ('DPR', '28'),
    ('DFM', '28'),
    ('DCU', '28'),
    ('MP3', '30'),
    ('M3U', '31'),
    ('WAV', '30'),
    ('MID', '30'),
    ('OGG', '30'),
    ('AVI', '32'),
    ('MPE', '32'),
    ('MPG', '32'),
    ('MPEG', '32'),
    ('BMP', '34'),
    ('GIF', '36'),
    ('JPG', '33'),
    ('PCX', '33'),
    ('ICO', '33'),
    ('INI', '20'),
    ('REG', '35'),
    ('PNG', '36'),
    ('MOV', '37'),
    ('XML', '21'),
    ('PDF', '38'));

Function IsDirectory(Item:TFileRecord):Boolean;
Function IsRoot(Path:String):Boolean;

Function GetFormattedName(Name:String):String;
Function GetFormattedExt(Ext:String):String;
Function GetFormattedSize(Size:Integer; SizeWidth:Byte;
UseSeparators:Boolean):String;
Function GetFormattedDateTime(DateTime:TDateTime):String;
Function GetFormattedAttr(Attr:Integer):String;
Procedure GetFormattedItem(Item:TFileRecord; Var
FormattedItem:TFileFormattedRecord);

Procedure GetItemByFileName(FileName:String; Var Item:TFileRecord);

```

```

Function GetItemImageIndex(Item:TFileRecord):Integer;

Function GetCompactSize(Size:Int64):String;

Function GetFullName(Item:TFileRecord):String;

implementation

Constructor TFiles.Init;
Begin
  slDirs:=TStringList.Create;
  slFiles:=TStringList.Create;
  ItemsCount:=0;
  Tag:=0;
End;

Destructor TFiles.Done;
Begin
  slDirs.Free;
  slFiles.Free;
End;

Function TFiles.Add(Item:TFileRecord):Boolean;
Begin
  Result:=False;
  If ItemsCount>=MaxFilesCount Then Exit;
  Result:=True;
  Item.ID:=ItemsCount;
  Items[ItemsCount]:=Item;
  ItemsID[ItemsCount]:=ItemsCount;
  Inc(ItemsCount);
End;

Procedure TFiles.Clear;
Begin
  slDirs.Clear;
  slFiles.Clear;
  ItemsCount:=0;
End;

Procedure TFiles.GetItem(ItemNo:Integer; Var Item:TFileRecord);
Begin
  If ((ItemNo<0) Or (ItemNo>=ItemsCount)) Then Exit;
  Item:=Items[ItemsID[ItemNo]];
End;

Procedure TFiles.FinishSort(Ascending, DirAscending:Boolean);
Var
  i:Integer;
Begin
  slDirs.Sort;
  slFiles.Sort;

  If slDirs.Count>0 Then
    For i:=0 To slDirs.Count-1 Do
      If DirAscending Then
        ItemsID[i]:=TRecordID(slDirs.Objects[i]).ID
      Else
        ItemsID[i]:=TRecordID(slDirs.Objects[slDirs.Count-i-1]).ID;
  If slFiles.Count>0 Then
    For i:=slDirs.Count To slFiles.Count-1+slDirs.Count Do
      If Ascending Then
        ItemsID[i]:=TRecordID(slFiles.Objects[i-slDirs.Count]).ID
      Else
        ItemsID[i]:=TRecordID(slFiles.Objects[slFiles.Count+slDirs.Count-i-1]).ID;
End;

```

```

Procedure TFiles.SortByName(Ascending:Boolean);
Var
  Item:TFileRecord;
  i:Integer;
Begin
  If ItemsCount<=0 Then Exit;
  slDirs.Clear;
  slFiles.Clear;
  For i:=0 To ItemsCount-1 Do
  Begin
    Item:=Items[i];
    If IsDirectory(Item) Then
    Begin
      slDirs.Add(GetFormattedName(Item.Name));
      DirsID[slDirs.Count-1].ID:=i;
      slDirs.Objects[slDirs.Count-1]:=TObject(DirsID[slDirs.Count-1]);
    End
    Else Begin
      slFiles.Add(GetFormattedName(Item.Name));
      FilesID[slFiles.Count-1].ID:=i;
      slFiles.Objects[slFiles.Count-1]:=TObject(FilesID[slFiles.Count-1]);
    End;
  End;
  FinishSort(Ascending, Ascending);
End;

Procedure TFiles.SortByExt(Ascending:Boolean);
Var
  Item:TFileRecord;
  i:Integer;
Begin
  If ItemsCount<=0 Then Exit;
  slDirs.Clear;
  slFiles.Clear;
  For i:=0 To ItemsCount-1 Do
  Begin
    Item:=Items[i];
    If IsDirectory(Item) Then
    Begin
      slDirs.Add(GetFormattedName(Item.Name));
      DirsID[slDirs.Count-1].ID:=i;
      slDirs.Objects[slDirs.Count-1]:=TObject(DirsID[slDirs.Count-1]);
    End
    Else Begin
      slFiles.Add(GetFormattedExt(Item.Ext));
      FilesID[slFiles.Count-1].ID:=i;
      slFiles.Objects[slFiles.Count-1]:=TObject(FilesID[slFiles.Count-1]);
    End;
  End;
  FinishSort(Ascending, True);
End;

Procedure TFiles.SortBySize(Ascending:Boolean);
Var
  Item:TFileRecord;
  i:Integer;
  TmpStr:String;
Begin
  If ItemsCount<=0 Then Exit;
  slDirs.Clear;
  slFiles.Clear;
  For i:=0 To ItemsCount-1 Do
  Begin
    Item:=Items[i];
    If IsDirectory(Item) Then
    Begin
      slDirs.Add(GetFormattedName(Item.Name));
      DirsID[slDirs.Count-1].ID:=i;
    End
  End;

```

```

        slDirs.Objects[slDirs.Count-1]:=TObject(DirsID[slDirs.Count-1]);
    End
    Else Begin
        TmpStr:=GetFormattedSize(Item.Size, 10, False);
        slFiles.Add(TmpStr);
        FilesID[slFiles.Count-1].ID:=i;
        slFiles.Objects[slFiles.Count-1]:=TObject(FilesID[slFiles.Count-1]);
    End;
    End;
    FinishSort(Ascending, True);
End;

Procedure TFiles.SortByDateTime(Ascending:Boolean);
Var
    Item:TFileRecord;
    i:Integer;
    TmpStr:String;
Begin
    If ItemsCount<=0 Then Exit;
    slDirs.Clear;
    slFiles.Clear;
    For i:=0 To ItemsCount-1 Do
        Begin
            Item:=Items[i];
            If IsDirectory(Item) Then
                Begin
                    slDirs.Add(GetFormattedName(Item.Name));
                    DirsID[slDirs.Count-1].ID:=i;
                    slDirs.Objects[slDirs.Count-1]:=TObject(DirsID[slDirs.Count-1]);
                End
            Else Begin
                Str(Item.DateTime:16:10, TmpStr);
                slFiles.Add(TmpStr);
                FilesID[slFiles.Count-1].ID:=i;
                slFiles.Objects[slFiles.Count-1]:=TObject(FilesID[slFiles.Count-1]);
            End;
        End;
    FinishSort(Ascending, True);
End;

Procedure TFiles.SortByAttr(Ascending:Boolean);
Var
    Item:TFileRecord;
    i:Integer;
Begin
    If ItemsCount<=0 Then Exit;
    slDirs.Clear;
    slFiles.Clear;
    For i:=0 To ItemsCount-1 Do
        Begin
            Item:=Items[i];
            If IsDirectory(Item) Then
                Begin
                    slDirs.Add(GetFormattedName(Item.Name));
                    DirsID[slDirs.Count-1].ID:=i;
                    slDirs.Objects[slDirs.Count-1]:=TObject(DirsID[slDirs.Count-1]);
                End
            Else Begin
                slFiles.Add(GetFormattedAttr(Item.Attr));
                FilesID[slFiles.Count-1].ID:=i;
                slFiles.Objects[slFiles.Count-1]:=TObject(FilesID[slFiles.Count-1]);
            End;
        End;
    FinishSort(Ascending, True);
End;

Procedure TFiles.GetInfo(Var TotalDirs, TotalFiles, CheckedDirs,
CheckedFiles:Integer; Var TotalSize, CheckedSize:Int64);
Var

```

```

    i:Integer;
Begin
    TotalDirs:=0;
    TotalFiles:=0;
    CheckedDirs:=0;
    CheckedFiles:=0;
    TotalSize:=0;
    CheckedSize:=0;

    If ItemsCount<=0 Then Exit;
    For i:=0 To ItemsCount-1 Do
    Begin
        If IsDirectory(Items[i]) Then
        Begin
            Inc(TotalDirs);
            If Items[i].Checked Then Inc(CheckedDirs);
        End
        Else Begin
            Inc(TotalFiles);
            If Items[i].Checked Then Inc(CheckedFiles);
        End;
        If Items[i].Checked Then CheckedSize:=CheckedSize+Items[i].Size;
        TotalSize:=TotalSize+Items[i].Size;
    End;
End;

Function IsDirectory(Item:TFileRecord):Boolean;
Begin
    If ((Item.Attr And faDirectory)>0) Then Result:=True Else Result:=False;
End;

Function IsRoot(Path:String):Boolean;
Begin
    Result:=(Path=IncludeTrailingBackslash(ExtractFileDrive(Path)));
End;

Function GetFormattedName(Name:String):String;
Begin
    Result:=Name;
End;

Function GetFormattedExt(Ext:String):String;
Begin
    Result:=Ext;
End;

Function GetFormattedSize(Size:Integer; SizeWidth:Byte;
UseSeparators:Boolean):String;
Var
    i, i3:Integer;
    Res:String;
Begin
    Str(Size:SizeWidth, Result);
    If Not(UseSeparators) Then Exit;
    If Length(Result)<=3 Then Exit;
    i3:=0;
    Res:=Result;
    Result:='';
    For i:=Length(Res) DownTo 1 Do
    Begin
        If i3=3 Then
        Begin
            Result:=ConstSizeSeparator+Result;
            i3:=0;
        End;
        Result:=Res[i]+Result;
        Inc(i3);
    End;
End;

```

```

End;

Function GetFormattedDateTime (DateTime:TDateTime) :String;
Begin
  Result:=DateTimeToStr (DateTime);
End;

Function GetFormattedAttr (Attr:Integer) :String;
Begin
  Result:='----';
  If ((Attr And faReadOnly)>0) Then Result[1]:=ConstReadOnly;
  If ((Attr And faArchive)>0) Then Result[2]:=ConstArchive;
  If ((Attr And faHidden)>0) Then Result[3]:=ConstHidden;
  If ((Attr And faSysFile)>0) Then Result[4]:=ConstSystem;
End;

Procedure GetFormattedItem (Item:TFileRecord; Var
FormattedItem:TFileFormattedRecord);
Begin
  FormattedItem.Name:=GetFormattedName (Item.Name);
  FormattedItem.Ext:=GetFormattedExt (Item.Ext);
  If IsDirectory (Item) Then
  Begin
    FormattedItem.Size:=ConstDirectory;
  End
  Else Begin
    FormattedItem.Size:=GetFormattedSize (Item.Size, 1, True);
  End;
  FormattedItem.DateTime:=GetFormattedDateTime (Item.DateTime);
  FormattedItem.Attr:=GetFormattedAttr (Item.Attr);
End;

Procedure GetItemByFileName (FileName:String; Var Item:TFileRecord);
Var
  F:File;
Begin
  FillChar (Item, SizeOf (Item), 0);
  If FileExists (FileName) Then
  Begin
    Item.Name:=ExtractFileName (FileName);
    Item.Ext:=ExtractFileExt (FileName);
    If Item.Ext<>' ' Then
    Begin
      Delete (Item.Name, Length (Item.Name)-Length (Item.Ext)+1, Length (Item.Ext));
      Delete (Item.Ext, 1, 1);
    End;
    Item.DateTime:=FileDateToDateTime (FileAge (FileName));
    Item.Attr:=FileGetAttr (FileName);

    {$I-}
    AssignFile (F, FileName);
    Reset (F, 1);
    If IOResult=0 Then
      Item.Size:=FileSize (F)
    Else
      Item.Size:=0;
    Close (F);
    {$I+}
    Exit;
  End;

  Item.Name:=ExtractFileName (FileName);
  Delete (Item.Name, 1, 1);
  Delete (Item.Name, Length (Item.Name), 1);
  If DirectoryExists (ExtractFilePath (FileName)+Item.Name) Then
  Begin
    Item.Ext:='';
    Item.Size:=0;
    Item.DateTime:=0;
  End;

```

```

    Item.Attr:=faDirectory;
End
Else Begin
    Item.Name:='';
End;
End;

Function GetItemImageIndex(Item:TFileRecord):Integer;
Var
    i:Integer;
Begin
    If Item.Name=ConstDirUp Then
        Begin
            Result:=10;
            Exit;
        End;
    If IsDirectory(Item) Then
        Begin
            Result:=11;
            Exit;
        End;
    Item.Ext:=UpperCase(Item.Ext);
    For i:=0 To MaxFileTypes-1 Do
        Begin
            If Item.Ext=Images[i, 0] Then
                Begin
                    Result:=StrToInt(Images[i, 1]);
                    Exit;
                End;
            End;
        End;
    Result:=19;
End;

Function GetCompactSize(Size:Int64):String;
Begin
    If Size<=ConstBytesLimit Then
        Begin
            Result:=GetFormattedSize(Size,1, True)+' '+ConstBytes;
            Exit;
        End;
    If Size<=ConstKBytesLimit Then
        Begin
            Result:=GetFormattedSize((Size Div 1024), 1, True)+' '+ConstKBytes;
            Exit;
        End;
    Result:=GetFormattedSize((Size Div (1024*1024)), 1, True)+' '+ConstMBytes;
End;

Function GetFullName(Item:TFileRecord):String;
Begin
    If Item.Ext='' Then
        Result:=Item.Name
    Else
        Result:=Item.Name+'.'+Item.Ext;
End;

end.

```

fmAnyMessageU.pas - повідомлення

```
unit fmAnyMessageU;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls;

type
  TFileAnyMessage = class(TForm)
    lbInformation: TLabel;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FileAnyMessage: TFileAnyMessage;

Procedure ShowAnyMessage(Caption, Information:String);
Procedure HideAnyMessage;

implementation

{$R *.DFM}

Procedure ShowAnyMessage(Caption, Information:String);
Begin
  FileAnyMessage.Caption:=Caption;
  FileAnyMessage.lbInformation.Caption:=Information;
  FileAnyMessage.Show;
  Application.ProcessMessages;
End;

Procedure HideAnyMessage;
Begin
  FileAnyMessage.Hide;
End;

end.
```

fmErrorDriveU.pas - обработка ошибок

```

unit fmErrorDriveU;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Buttons, FileCtrl;

type
  TFileError = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    dcbxDrive: TDriveComboBox;
    bbOk: TBitBtn;
    procedure bbOkClick(Sender: TObject);
    procedure FormShow(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FileError: TFileError;

Function ChooseNewDrive(Var Drive:Char):Boolean;

implementation

uses StrConsts;

{$R *.DFM}

Function ChooseNewDrive(Var Drive:Char):Boolean;
Begin
  FileError.dcbxDrive.Drive:=Drive;
  If FileError.ShowModal=mrOk Then
  Begin
    Drive:=FileError.dcbxDrive.Drive;
    Result:=True;
  End
  Else
    Result:=False;
End;

procedure TFileError.bbOkClick(Sender: TObject);
begin
  Hide;
end;

procedure TFileError.FormShow(Sender: TObject);
begin
  dcbxDrive.SetFocus;
end;

end.

```

StrConsts.pas - панель стану

```
unit StrConsts;

interface
Uses
  Graphics;

Const
  ConstDirUp='..';
  ConstDirectory:String='<ПАПКА>';
  ConstReadOnly:Char='r';
  ConstArchive:Char='a';
  ConstHidden:Char='h';
  ConstSystem:Char='s';

  ConstNoSort:Char='•';
  ConstSortAscending:Char='+';
  ConstSortDescending:Char='-';

  ConstDirLeftBracket:Char='[';
  ConstDirRightBracket:Char=']';

  ConstBytesLimit:Integer=99999;
  ConstKBytesLimit:Integer=99999*1024;
  ConstBytes:String='байт';
  ConstKBytes:String='Кбайт';
  ConstMBytes:String='Мбайт';

  ConstSizeSeparator:String=' ';

  ConstUnCheckedImageIndex:Integer=-1;
  ConstCheckedImageIndex:Integer=9;

  ConstTypeDirectory:String='папка';
  ConstTypeFile:String='файл';

  ConstNotepadFile:String='notepad.exe';

  ConstLabelActiveColor:Integer=clNavy;
  ConstLabelNonActiveColor:Integer=clGray;

implementation

end.
```

About.pas - довідка

```
unit about;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, jpeg, ExtCtrls;

type
  TFmAbout = class(TForm)
    Memo1: TMemo;
    Button1: TButton;
    Image1: TImage;
    procedure FormCreate(Sender: TObject);
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FmAbout: TFmAbout;

implementation

{$R *.dfm}

procedure TFmAbout.FormCreate(Sender: TObject);
begin
  Memo1.Clear;
  Memo1.Lines.Add('МАГІСТЕРСЬКА РОБОТА');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('на тему:');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('Дослідження та програмна реалізація системи менеджменту файлів  
ОС Windows 11/Windows Server 2022');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('Керівник: Доренський О.П. ');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('Розробив: студент Дудченко Анастасія Василівна ');
  Memo1.Lines.Add('гр. КН-22М-1');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('м. Кропивницький 2023');
  Memo1.Lines.Add('');
end;

procedure TFmAbout.Button1Click(Sender: TObject);
begin
  FmAbout.Close;
end;
end.
```