

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

”Допущено до захисту”

Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор

Олексій СМІРНОВ

“ ___ ” _____ 2021 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему

**“Дослідження та програмна реалізація системи IP-телефонії з
забезпеченням конфіденційності”**

Виконав здобувач вищої освіти

II курсу, групи КН-20МЗ

ОПП «Комп’ютерні науки»

спеціальності 122 «Комп’ютерні науки»

_____ Піскова О.А.

« ___ » _____ 2021 р.

Керівник проекту

кандидат технічних наук

_____ Тетяна СМІРНОВА

« ___ » _____ 2021 р.

Рецензент _____

Центральноукраїнський національний технічний університет

Факультет Механіко-технологічний

Центр Заочної та дистанційної освіти

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 2021 року

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

Пісковій Олені Андріївні

(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження та програмна реалізація системи IP-телефонії з забезпеченням конфіденційності*

2. Керівник роботи *Смірнова Тетяна Віталіївна, канд. техн. наук*

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

затверджені наказом вищого навчального закладу № 40-13 від 02.08.2021 року

3. Строк подання студентом роботи до захисту *10.12.2021 р.*

4. Мета та завдання випускної кваліфікаційної роботи: *Метою розробки є дослідження та програмна реалізація системи IP-телефонії з забезпеченням конфіденційності*

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

<i>1. Призначення та область використання.</i>	<i>7. Економічна ефективність розробленої програми.</i>
<i>2. Перегляд аналогічних існуючих систем.</i>	<i>8. Заходи з охорони праці та техніки безпеки</i>
<i>3. Опис і обґрунтування проектних рішень.</i>	<i>9. Висновки.</i>
<i>4. Етапи програмування системи.</i>	
<i>5. Впровадження системи в промислову експлуатацію</i>	
<i>6. Наукова новизна</i>	
6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)	
<i>Наукова новизна</i>	<i>1 аркуш</i>
<i>Структурна схема системи</i>	<i>1 аркуш</i>
<i>Функціональна схема системи</i>	<i>1 аркуш</i>
<i>Діаграма процесів</i>	<i>1 аркуш</i>
<i>Блок-схема алгоритму роботи додатку</i>	<i>2 аркуша</i>
<i>Показники економічної ефективності</i>	<i>1 аркуш</i>

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Савеленко Г.В.	05.10.2021	14.11.2021
Охорона праці	Оришака О.В.	06.10.2021	16.11.2021

7. Дата видачі завдання « 6 » вересня 2021 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	10.10.2021 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2021 р.	
3.	Розробка моделі компонента	20.10.2021 р.	
4.	Розробка структур даних	25.10.2021 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2021 р.	
6.	Програмування алгоритмів	10.11.2021 р.	
7.	Розрахунок економічної ефективності	13.11.2021 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2021 р.	
9.	Оформлення ПЗ	17.11.2021 р.	
10.	Попередній захист роботи	10.12.2021 р.	

Дата видачі завдання
« 6 » вересня 2021 р.

Підпис керівника

_____ (прізвище та ініціали)

Завдання прийнято до виконання
« 6 » вересня 2021 р.

Підпис здобувача

_____ (прізвище та ініціали)

АНОТАЦІЯ

Піскова О.А. Дослідження та програмна реалізація системи IP-телефонії з забезпеченням конфіденційності. 122 Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2021.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи IP-телефонії з забезпеченням конфіденційності.

Метою розробки є дослідження та програмна реалізація системи IP-телефонії з забезпеченням конфіденційності.

Об'єктом дослідження є процес IP-телефонії з забезпеченням конфіденційності.

Предметом дослідження є методи IP-телефонії з забезпеченням конфіденційності.

Методи дослідження базуються на методах теорії передачі мультимедійних даних, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи IP-телефонії з забезпеченням конфіденційності.

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 XP/Vista/7/8/10.

Програму розроблено в середовищі RAD Studio Delphi 10.

Ключові слова: Комп'ютерні науки, IP-телефонія

ABSTRACT

Piskova O.A. Research and software implementation of IP telephony system with confidentiality. 122 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2021

In this final qualification work on the second (master's) level of higher education the software which is intended for system of IP -telephony with maintenance of confidentiality is developed.

The purpose of the development is to research and software implementation of the IP-telephony system with the provision of confidentiality.

The object of research is the process of IP telephony with the provision of confidentiality.

The subject of the research is the methods of IP-telephony with ensuring confidentiality.

Research methods are based on the methods of the theory of multimedia data transmission, methods of mathematical statistics, methods of software development.

The result is a software implementation of an IP telephony system with privacy.

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.

Developed user-friendly interface. Instructions for working with software are given.

The program can be used on an IBM PC with Windows XP / Vista / 7/8/10.

The program is developed in the environment of RAD Studio Delphi 10.

Keywords: Computer Science, IP Telephony

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП 5	
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ.....	8
1.1 Призначення системи.....	8
1.2 Область застосування.....	9
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	14
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	14
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	19
2.3 Розгорнута постановка завдання	25
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	27
3.1 Опис функціонування системи.....	27
3.2 Розробка структурної схеми	41
3.3 Розробка функціональної схеми.....	45
3.4 Розробка діаграми процесів.....	47
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ ...	48
4.1 Розробка блок-схем та опис алгоритмів функціонування системи	48
4.2 Захист розробленого програмного забезпечення	60
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ.....	66
6 НАУКОВА НОВИЗНА	75

ВКРМ-122.21.0094.00.00.ПЗ

Вим.	Арк.	№ докум.	Підп.	Дата				
Розроб.		Піскова О.А.			Дослідження та програмна реалізація системи IP-телефонії з забезпеченням конфіденційності	Лім.	Аркуш	Аркушів
Перев.		Смірнова Т.В.				М	1	112
Н.контр.		Гермак В.С.			ЦНТУ КН-20МЗ			
Затв.		Смірнов О.А.						

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	76
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.	76
7.2 Розрахунок трудомісткості розробки програмної продукції	78
7.3 Визначення чисельності виконавців і планового фонду зарплати	80
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника	85
7.5 Визначення собівартості розробки та ціни програмної продукції.	89
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	91
7.7 Визначення експлуатаційних витрат.....	91
7.8 Визначення економічної ефективності програмної продукції.....	93
7.9 Висновок.	95
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	96
8.1 Вступ.....	96
8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером	97
8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста ..	98
8.4 Розробка заходів з умов поліпшення охорони праці.....	101
8.5 Розрахункова частина	102
8.6 Висновки до розділу.....	104
9 ОСНОВНІ ВИСНОВКИ.....	105
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	107

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

БД	– база даних
КМЗ	– корпоративна мережа зв'язку
ЛОМ	– локальна обчислювальна мережа
ОЗП	– оперативно-запам'ятовувальний пристрій
ПАТМ	– пристрій автоматизованої телефонної мережі
ПЗ	– програмне забезпечення
ПП	– програмний продукт
СПД	– система передачі даних
СУБД	– система управління БД
ТфОП	– телефонний оператор
CNG	– Comfort Noise Generator. Генератор комфортного шуму
DSP	– Digital Signal Processor. Процесор цифрової обробки сигналів
DTMF	– Dual Tone Multi-Frequency. Багаточастотна система кодування цифр номера
GK	– Gatekeeper. Воратар . Виконує функції керування зоною мережі H.323
GW	– Gateway. Шлюз. Апаратно-програмний комплекс, що забезпечує обмін даними між мережами різних типів
H.323	– Рекомендація ІТУ-Т, що визначає системи мультимедійного зв'язку в мережах з пакетною комутацією
H.248	– Протокол керування транспортним шлюзом
ІР	– Internet Protocol. Протокол міжмережної взаємодії
LPC	Linear Prediction Coding. Кодування з лінійним проорокуванням
MCU	– Multipoint Control Unit. Пристрій керування конференцією
MG	– Media Gateway. Транспортний шлюз
MGCP	– Media Gateway Control Protocol. Протокол керування шлюзами
MP	– Multipoint processor. Процесор для обробки інформації користувачів при централізованих конференціях

- OSI – Open System Interconnection. Взаємодія відкритих систем
- PPP – Point-to-Point Protocol. Протокол двостороннього зв'язку
- RADIUS – Remote Authentication Dial-In User Service. Протокол автентифікації й авторизації абонентів, а також обліку обсягу наданих їм послуг
- RAS – Registration Admission and Status. Протокол взаємодії термінального встаткування з gatekeeper. Входить у сімейство протоколів H.323
- RSVP – Resource Reservation Protocol. Протокол резервування ресурсів
- RTCP – Real-time Transport Control Protocol. Протокол контролю транспортування інформації в реальному часі
- SIP – Session Initiation Protocol. Протокол ініціювання сеансів зв'язку
- TAPI – Telephony Applications Programming Interface. Інтерфейс для програмування телефонних додатків
- TCP – Transmission Control Protocol. Протокол керування передачею (даних) Основний транспортний протокол у стеці протоколів TCP/IP. Встановлює зв'язок між двома ПК й організує обмін даними в дуплексному режимі
- TCP/IP – Transmission Control Protocol/Internet Protocol. Стек протоколів, що забезпечують організацію зв'язку між комп'ютерами в мережі Інтернет
- UDP – User Datagram Protocol. Протокол передачі дейтаграмм користувача. Подібно TCP, використовує для доставки даних протокол IP. На відміну від TCP/IP, передбачає обмін дейтаграммами без підтвердження
- VoIP – Voice over Internet Protocol. Технологія, що дозволяє використовувати IP-мережу для передачі мовної інформації

ВСТУП

Актуальність теми. З початком ери комп'ютеризації стали бурхливо розвиватися системи передачі даних (СПД) на базі IP [1-3]. Організувавши високошвидкісний доступ до СПД по наземних каналах або з використанням доступної бездротової технології, оператор може надати клієнтові й послуги телефонного зв'язку – досить лише підключити телефонні апарати або персональну автоматичну телефону мережу (ПАТМ) клієнта через мережу IP до платформи VoIP оператора. Аналогічні підходи застосовні й при побудові корпоративної мережі зв'язку (КМЗ) [5]. Шлях побудови корпоративної мережі зв'язку на базі VoIP припускає відмову від традиційної комутації (тобто від комутації каналів) за допомогою ПАТМ і впровадження IP-PBX (IP-ПАТМ), які є новим поколінням систем зв'язку, орієнтованим на VoIP [1-10]. Замість IP-ПАТМ у КМЗ для комутації трафіку VoIP можна скористатися послугами операторів зв'язку Hosted IP-PBX (віртуальна IP-ПАТМ) або IP-Centrex (оренда комутаційної ємності). Більша частина середніх і великих підприємств має офіси в декількох районах і містах. Організація зв'язку із центральним офісом коштує чималих грошей. Тим часом діяльність всіх перерахованих компаній і підприємств не можна уявити без доступу співробітників до мережі Internet і/або корпоративної обчислювальної мережі. Однак однією з умов придатності приміщення для розміщення філії є наявність мережі передачі даних і телефонної мережі. Тільки от чи не так уже необхідно сьогодні окреме підключення до ТфОП? Або, скажемо інакше, чи багато потрібно далеко не дешевих телефонних ліній і чи потрібна окрема ПАТМ, яку доводиться обслуговувати на місці? Використання у філіях електронної пошти, корпоративного електронного документообігу й внутрішніх ресурсів Web (Intranet), довідкових систем і баз даних, систем обліку товарно-фінансових потоків і керування ресурсами підприємства вимагають організації надійного доступу до корпоративної мережі

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
<i>Вим.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		5

по виділених каналах або через мережу передачі даних за допомогою VPN. Але якщо такий доступ реалізований, то ці ж канали придатні й для телефонного зв'язку – досить реалізувати підтримку технологій VoIP. Завдяки шлюзам VoIP сьогодні можна створювати налагоджену КМЗ з можливістю доступу до будь-якого співробітника без виходу до ТфОП. Для цього абоненти філій підключаються до ПАТМ центрального офісу (яка, як правило, уже є) за допомогою абонентських місць побудованих на основі шлюзів VoIP [9-10].

Основна перевага використання технології VoIP у корпоративній мережі зв'язку – це впровадження з метою економії на капітальних і операційних витратах, пов'язаних з побудовою (на підготовці приміщень і трас, покупці встаткування і його монтажі, будівництві ліній зв'язку, придбанні портової й номерної ємності операторів і т.і.) і експлуатацією (на зарплаті власному сервісному персоналу, сервісних послугах сторонніх організацій, оренді портової й номерної ємності, витратах на трафік і т.і.).

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи IP-телефонії з забезпеченням конфіденційності.

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

- Огляд існуючих систем IP-телефонії з забезпеченням конфіденційності.
- Дослідження системи IP-телефонії з забезпеченням конфіденційності.
- Програмна реалізація системи IP-телефонії з забезпеченням конфіденційності.

Об'єктом дослідження є процес IP-телефонії з забезпеченням конфіденційності.

Предметом дослідження є методи IP-телефонії з забезпеченням конфіденційності.

Методи дослідження базуються на методах теорії передачі мультимедійних даних, методах математичної статистики, методах розробки програмного забезпечення.

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

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

- Удосконалено метод IP-телефонії з забезпеченням конфіденційності.
- Розроблено вітчизняний продукт IP-телефонії з забезпеченням конфіденційності, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі IP-телефонії з забезпеченням конфіденційності.

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 LV Науково-технічна конференція здобувачів вищої освіти «Наука – виробництву», 2021, основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №12.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи IP-телефонії з забезпеченням конфіденційності, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Загальноприйнято, що раціоналізація інформаційного процесу з поширенням на нього елементів виробничої діяльності (нормування, технологія) повинна підвищити ефективність управлінської праці. Одним з основних показників ефективності роботи підприємства є його продуктивність: якість, кількість і швидкість обробки інформації. Будь-яка організація обробляє інформацію для вироблення двох видів «продукції»: інформації (даних, документів, мовної інформації) і рішень (оперативних і стратегічних). Організація одержує початкову (вхідну) інформацію в різних видах: документи, що містять інформацію у вигляді слів і цифр; мовну інформацію з телефону; дані від ЕОМ в електронній формі. Кінцева (вихідна) інформація отримується в таких же видах. Виробничий цикл організації може включати перекомпонування інформації, об'єднання даної інформації з іншою, нагромадження інформації. Найбільш прості задачі утворюють клас цілком формалізованих (або добре структурованих) процедур, виконання яких, крім витрат часу, складності для виконавців не представляє. Ці задачі легко стандартизуються і програмуються. До таких задач відносяться: облік і контроль, оформлення документів, їхнє тиражування і розсилання і т.п.

Основою вирішення задач залишаються творчий потенціал людини і різні атрибути її діяльності (інформованість, кваліфікація, талант, інтуїція і т.п.).

Критерієм продуктивності роботи інформаційних систем є оперативність і своєчасність інформаційної обробки, а також підтримка високої пропускну здатності організації з мінімальною кількістю збоїв і помилок.

Таким чином, в організації наявні виробничі задачі і виконавці цих задач. Залишається визначити необхідні організації технологічні процеси автоматизації.

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

В зв'язку з тим, що основою стабільної та стійкої роботи любого підприємства є система керування, приділимо основну увагу автоматизації саме цієї системи. Основною частиною системи керування є підсистема зв'язку. У якості корпоративної мережі зв'язку пропонується застосування мережі на основі IP-телефонії. Інтенсивний розвиток інформаційних технологій і технічний прогрес сприяють тому, що великі можливості IP-телефонії стають усе більше доступними для кінцевих користувачів. Зараз застосування IP-телефонії економічно вигідно не тільки як можливість скорочення витрат на оплату міжміських і міжнародних переговорів, але і як заміна корпоративної АТМ – тобто для побудови повної захищеної системи зв'язку корпорації.

1.2 Область застосування

Розглянемо *основні переваги* IP-телефонії в порівнянні із традиційною телефонією [9]:

– Економічна ефективність. В IP-телефонії голос і дані передаються по тому самому фізичному носію. Це означає, що загальну кількість фізичних ліній зв'язку в кабельній системі може бути значно скорочено. Крім того, підвищується ефективність використання орендованих глобальних інформаційних каналів за рахунок використання єдиного каналу для передачі голосового трафіка й трафіка даних, при цьому можлива значна економія на оплаті міжміських телефонних розмов між різними відділеннями компанії.

– Масштабованість і легкість розширення. За допомогою засобів IP-телефонії можна побудувати мережу, що охоплює від декількох десятків користувачів у мережі малого підприємства до декількох десятків тисяч абонентів у розподіленій мережі великої корпорації. Для збільшення кількості користувачів телефонії досить просто встановити необхідну кількість IP-телефонів.

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

– IP-телефонія базується на відкритих міжнародних стандартах і протоколах. Це означає, що її легше впроваджувати й обслуговувати. Немає необхідності утримувати окремий штат телефоністів або звертатися в спеціалізовані сервісні фірми. Керування навіть великою розподіленою мережею здійснюється централізовано системним адміністратором. Крім того, система, побудована на основі IP-телефонії, менш піддана "моральному старінню", – з появою нових функцій і можливостей вони легко можуть бути реалізовані у вже існуючій системі.

– Можливість захисту інвестицій, уже вкладених у телефонію. Система IP-телефонії легко інтегрується з існуючими АТМ і телефонними мережами загального користування. Можна передбачити поступовий етапний перехід від традиційної до IP-телефонії.

– Забезпечення відказостійкості. До мережі IP-телефонії застосовні всі ті ж методи й технології забезпечення відказостійкості, що й у звичайній комп'ютерній мережі. Крім того, у кожного виробника існують спеціальні методи по збереженню резервного телефонного зв'язку у випадку втрати повідомлення між віддаленим офісом і центром комутації телефонних викликів.

– Широке коло сучасних послуг зв'язку. У системі IP-телефонії можна реалізувати всі функції сучасних телефонних станцій, і, крім того, ряд нових можливостей, таких як: голосова пошта, уніфікована обробка голосових/факсових повідомлень і електронної пошти, система інтерактивних голосових меню, інтелектуальні центри обробки викликів, доступ до корпоративних директорій абонентів. На додаток до цього клієнти мають можливість створення власних додатків для IP-телефонії, що найбільше точно відповідають потребам користувачів.

Беручи до уваги все вищесказане, а також постійне зниження цін на комунікаційне встаткування й збільшення кількості якісних каналів передачі даних, можна стверджувати, що реалізацію корпоративної системи зв'язку

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

необхідно реалізовувати не на традиційній телефонії з використанням АТМ а на основі ІР-телефонії.

Характеристика підприємства

Як правило територіально корпорація розташована таким чином: головний офіс знаходиться у одному місті, а ряд дочірніх підприємств знаходяться у інших містах України та близького зарубіжжя (Росії та Білорусії). На території головного офісу виділена площа під склади, де зберігаються основні види товарів, які продаються клієнтам, тому менеджери можуть контролювати наявність матеріалу на складі, відвантаження та проводити інвентаризацію в безпосередній близькості від офісу. Завдяки цьому спрощена схема документообігу. Виробництво знаходиться як на території України, так і в інших країнах, при цьому структура організації дочірніх підприємств є єдиною.

Продукція надходить безпосередньо з виробництва і реалізується клієнтам, що зацікавлені у даному виді товару. Організація здійснює транспортування продукції від виробництва на базу, її складування, збереження і відвантаження клієнтам. Номенклатура реалізованих товарів складає кілька сотень найменувань і постійно збільшується.

Крупна корпорація, як правило, веде оптову торгівлю за безготівковий розрахунок. Задовольняючи потреби роздрібних торговців, підприємство оптової торгівлі одержує прибуток від торгової надбавки на товари, що реалізуються.

В зовнішній системі організація корпорації виконує роль виробника товарів та реалізатора клієнтам розроблених систем. Наочно це можна побачити з рисунок 1.1.

На чолі корпорації знаходиться генеральний директор, що вирішує в основному управлінські питання, а також питання стратегічного характеру. Він контролює діяльність усіх відділів. Також у його компетенції є питання руху фінансових потоків.

Регіональний директор знаходиться на чолі підпорядкованого дочірнього підприємства.

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

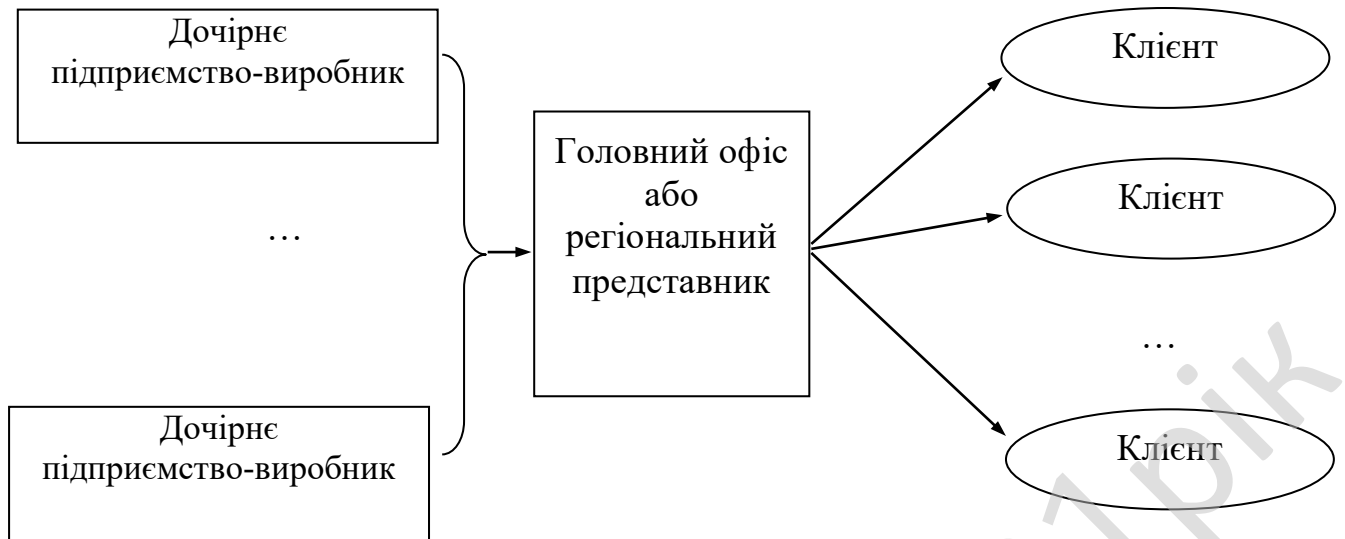


Рисунок 1.1 – Схема зовнішнього середовища корпорації

Закупівлею комплектуючих, для виробництва на промислових підприємствах займається відділ постачання. У його функції також входить пошук нових постачальників з більш вигідними умовами постачання. Цей відділ вирішує питання закупівлі щодо всього асортименту. Закупівля здійснюється на підставі заявок покупців, оформлених менеджерами.

До відділу збуту входять менеджери, що займаються безпосередньо клієнтами. Кожен менеджер працює з базою клієнтів фірми, збільшуючи її за рахунок пошуку нових клієнтів. Менеджер укладає договори, обговорює терміни, умови постачання й оплати товару; здійснює прийом замовлень від покупців.

Відділ бухгалтерії складається з трьох підвідділів: це головний бухгалтер, виписка, платежі і каса.

Головний бухгалтер веде бухгалтерський облік, робить баланс, різні зустрічні звірення, нараховує всі податки і вирішує питання, що виникають у відділах, які йому підпорядковуються.

Відділ «Платежі» припускає відправлення й одержання платежів з банку через спеціальну банківську програму. Це оплата по рахунках постачальників комплектуючих, сплата податків, щоденні одержання коштів на розрахунковий рахунок фірми від клієнтів й інші платежі.

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

Касир займається розподілом наявних засобів фірми (видача заробітної плати, виділення засобів на господарські витрати, видача коштів на відрядження і т.і.)

У відділ економічної безпеки входять: начальник охорони, що відповідає за безпеку об'єкта і пропускну систему; системний адміністратор, обов'язком якого є збереження інформаційної безпеки і забезпечення функціональності обчислювального комплексу.

Автоматизована система призначена для вирішення задач забезпечення захищеного зв'язку в корпорації.

Далі розглянемо докладніше існуючі системи, та обґрунтуємо вибір принципу розробки та методики побудови системи.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи IP-телефонії з забезпеченням конфіденційності, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

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

Основні алгоритми кодування мови, які використовуються в IP-телефонії

Кодери вихідної інформації (вокодери) і гібридні алгоритми. Багато методів кодування використовують особливості людської мови, зв'язані з будовою голосового апарата. Кодери, у яких реалізуються такі методи, називають кодерами вихідної інформації або вокодерами (voice coding).

Послідовність формованих звуків становить тонову мову. Якщо звук формується байдужості зв'язувань, тон у ньому відсутній, і послідовність таких звуків становить нетонову мову. Спектр тонового звуку може бути змодельований шляхом подачі спеціальним образом сформованого сигналу порушення на вхід цифрового фільтра з параметрами, обумовленими декількома дійсними коефіцієнтами. Спектр нетонових звуків – практично рівномірний, що обумовлено їхнім шумовим характером. У реальних мовних сигналах не всі звуки можна чітко розділити на тонові і нетонові, а доводиться мати справу з якимись перехідними варіантами, що утрудняє створення алгоритмів кодування, які забезпечують високу якість передачі мови при низькій швидкості передачі інформації. Описаний принцип кодування одержав назву LPC (Linear Prediction Coding – кодування з лінійним проорокуванням), оскільки центральним елементом моделі голосового тракту є лінійний фільтр. Алгоритми кодування форми сигналу засновані на наявності кореляційних зв'язків між відрахунками сигналу, які дають можливість лінійного проорокування. У сполученні з адаптивним квантуванням цей підхід дозволяє забезпечити гарну якість мови при швидкості

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

передачі біт порядку 24-32 Кбіт/с. LPC-кодери (вокодери) використовують просту математичну модель голосового тракту й дозволяють використовувати дуже низькі швидкості передачі інформації 1200-2400 біт/с, однак ціною «синтетичного» характеру мови.

Гібридні алгоритми кодування й алгоритми типу «аналіз шляхом синтезу» (ABS) являють собою спроби сполучити позитивні властивості двох описаних вище основних підходів і будувати ефективні схеми кодування з діапазоном швидкостей передачі біт 6-16Кбіт/с. Важлива відмінність кодерів такого типу полягає в тому, що в рамках цих алгоритмів немає необхідності ухвалювати рішення щодо типу відтвореного звуку (тонов або нетоновий), тому що передбачаються спеціальні міри для кодування сигналу помилки після проходження порушення через LPC-Фільтр [11-15].

Процесори цифрової обробки сигналів для мовних кодеків. Процесори DSP (цифрової обробки сигналів) мають архітектуру, оптимізовану для виконання операцій, які характерні для типових алгоритмів обробки сигналів. Архітектура процесорів DSP часто характеризується наявністю декількох обчислювальних блоків, що забезпечують виконання одночасних операцій в одному такті роботи процесора. Для завантаження обчислювальних блоків даними передбачається кілька шин передачі даних і багатопортова пам'ять даних. Для збільшення продуктивності пам'ять інструкцій і пам'ять даних розділені, а доступ до них здійснюється також по роздільних шинах. Для процесорів DSP характерне використання інструкцій збільшеної довжини, що містять поля для керування всіма обчислювальними блоками [11-15].

Алгоритми кодування

У першу чергу необхідно зрозуміти, якими критеріями потрібно керуватися при виборі кодека для використання в IP-телефонії [15-22].

Швидкість передачі, що передбачають наявні сьогодні вузькополосні кодеки, лежить у межах 1.2 – 64 Кбіт/с. Природно, що від цього параметра прямо залежить якість відтвореної мови. У рамках існуючих технологій якість ТфОП

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

(toll quality) неможливо забезпечити при швидкостях менш 5 Кбіт/с.

Придушення періодів мовчання (VAD, CNG, DTX). Детектор мовної активності (VAD) необхідний для визначення періодів часу, коли користувач говорить. Підтримка переривчастої передачі (DTX) дозволяє кодеку припинити передачу пакетів у той момент, коли VAD виявив період мовчання. Генератор комфортного шуму (CNG) служить для генерації фонового шуму.

Більшість вузькополосних кодеків обробляють мовну інформацію блоками, названими кадрами (frames), і тому необхідно здійснювати попередній аналіз обчислень, що впливають безпосередньо за обчисленнями в блоці, що вони в цей момент кодують. Розмір кадру важливий, тому що мінімальна теоретично досяжна затримка передачі інформації (алгоритмічна затримка) визначається сумою цього параметра й довжини буфера попереднього аналізу. У дійсності процесори цифрової обробки сигналів, які виконують алгоритм кодування, мають кінцеву продуктивність, так що реальна затримка сигналу більше теоретичної. До кадру, сформованому кодеком, додається безліч додаткової інформації – заголовки IP (20 байтів), UDP (8 байтів), RTP (12 байтів). Для кодеку із тривалістю кадру 30 мс послідовна передача таких кадрів по мережі привела б до передачі надлишкової інформації зі швидкістю 10.6 кбіт/с, що перевищує швидкість передачі мовної інформації в більшості вузькополосних кодеків.

Тому звичайно використовується пересилання декількох кадрів у пакеті, при цьому їхня кількість обмежена максимально припустимою затримкою. У більшості випадків в одному пакеті передається до 60 мс мовної інформації. Чим менше тривалість кадру, тим більше кадрів доводиться впаковувати в один пакет, тобто затримка визначається зовсім не довжиною кадру, а практично прийнятним обсягом корисного навантаження в пакеті. Крім того, кодеки з більшою довжиною кадру більше ефективні, тому що тут діє загальний принцип: чим довше спостерігається явище (мовний сигнал), тим краще воно може бути змодельоване.

ВКРМ-122.21.0094.00.00.ПЗ

Арк.

16

Вим.	Арк.	№ докум.	Підпис	Дата

Втрати пакетів є невід'ємним атрибутом IP-мереж. В зв'язку з тим, що пакети містять кадри, сформовані кодеком, то це викликає втрати кадрів. Але втрати пакетів і втрати кадрів не обов'язково прямо зв'язані між собою, тому що існують підходи, що дозволяють зменшити число загублених кадрів при даному числі загублених пакетів. Додаткова службова інформація розподіляється між декількома пакетами, так що при втраті деякого числа пакетів кадри можуть бути відновлені. Однак позитивний ефект від введення надмірності для боротьби із втратами пакетів не настільки легко досяжний, оскільки втрати в IP-мережах відбуваються пачками, тобто значно більш імовірно те, що буде загублено відразу кілька пакетів підряд, ніж те, що загублені пакети розподіляться в послідовності переданих пакетів по одному. Тому якщо застосовувати прості схеми введення надмірності, то в реальних умовах вони, хоча й збільшать обсяг надлишкової інформації, але, швидше за все, виявляться марними. Крім того, введення надмірності негативно позначається на затримці відтворення сигналу.

Огляд технічних засобів IP-телефонії

Невід'ємну частину процесу розробки складної інформаційної системи складає рішення таких ключових питань, як вибір складу обчислювальної техніки, визначення їхніх характеристик.

Критеріями вибору технічних засобів є: надійність функціонування системи; функціональна повнота системи; швидкодія; мінімізація витрат на вартість: апаратних засобів, прикладних систем, супроводу та розвитку системи.

Щоб проєктована корпоративна мережа зв'язку була ефективною необхідна дотримання наступних принципів створення системи: системність; гнучкість; стійкість; ефективність.

Відповідно до принципу системності, корпоративну мережу IP-зв'язку варто розглядати як системи, структура яких визначається функціональним призначенням. Принцип гнучкості означає пристосованість системи до можливих перебудов, завдяки модульності побудови всіх підсистем і стандартизації їхніх елементів. Принцип стійкості полягає в тому, що корпоративна мережа IP-зв'язку

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

повинна виконувати основні функції незалежно від впливу на неї внутрішніх і зовнішніх факторів. Це значить, що неполадки в окремих її частинах повинні легко усуватись, а працездатність системи швидко відновлювана.

Таблиця 2.1 – Обладнання для формування IP-телефонії

Найменування	Короткий опис	Приблизна ціна
IP-телефон	Повнофункціональний IP-телефон дає можливість користувачеві ініціювати й приймати дзвінки, взаємодіючи з корпоративною VoIP-системою або інфраструктурою постачальника послуг IP-телефонії.	\$120
VoIP шлюз	Повністю підтримуються рекомендації стандарту SIP 2.0 (RFC 3261) і дають можливість користувачеві ініціювати й приймати дзвінки, взаємодіючи з корпоративною VoIP-системою або інфраструктурою постачальника послуг IP-телефонії.	\$255
Proxy/ Маршрутизатор H.323-викликів	Proxy/маршрутизатор H.323-викликів являє собою Gatekeeper зі спеціалізованим програмним забезпеченням, що забезпечує для мереж IP-телефонії, роботу із протоколу H.323.	\$9800
Шлюз на базі інтелектуальної платформи	Шлюз на базі інтелектуальної платформи реалізує передачу мовного трафіку й факсимільної інформації з мереж з маршрутизацією пакетів IP по протоколі відповідно до Рекомендації ITU-T H.323 v2.	\$5800
Білінгова система	Білінгова система – автоматизована система розрахунків з абонентами за надані послуги. Керування й статистика може бути доступна з будь-якої точки мережі через Веб-інтерфейс.	\$1500

Ефективність корпоративної мережі IP-зв'язку варто розглядати як інтегральний показник рівня реалізації приведених вище принципів, віднесеного до витрат на створення й експлуатацію системи [9,10].

Функціонування корпоративної мережі IP-зв'язку може дати бажаний ефект за умови правильного розподілу функцій і навантаження між людиною і машинними засобами обробки інформації, ядром якого є комп'ютер. Вибір технічних засобів та програмного забезпечення взаємозв'язані. В першу чергу вони залежать від розв'язуваних задач, а також від фінансових можливостей. Враховується також наявний досвід проектування подібних систем.

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.

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

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 – використовувачі класичну реалізацію керування пам'яттю об'єктів.

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

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

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

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

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

– Підтримка Metal Driver GPU для macOS і iOS.

– Вбудований Fmxlinux.

– Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.

Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

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

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

– Додані оновлені драйвери для FireBird, PostgreSQL і SQLite.

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

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

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

RAD Studio 10.4 Короткий огляд:

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

– Зросла продуктивність розробки. Ріст продуктивності за рахунок миттєвої реакції підказок code completion у середовищі IDE. Краща сумісність із уже наявною кодовою базою, і спрощення програмування за рахунок уніфікованої архітектури керування пам'яттю. Швидке зв'язування даних і візуальних елементів за допомогою розширеної технології Visual LiveBindings з підвищеною швидкістю. 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,

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

навігації й інших сервісів в окремому процесі. Це значить, що code completion і Code Insight одержать більш точні результати без блокування IDE. 10.4 забезпечує набагато більш високу продуктивність розроблювачів, які працюють із більшими проектами, що містять мільйони рядків коду.

Delphi Custom Managed Records

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

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

Єдине керування пам'яттю

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

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

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

Розширена підтримка бібліотек C++

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

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

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

Win 64-відладник і збирач для C++

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

Підвищення якості й швидкодії інструментів

- Велика кількість поліпшень STL від Dinkumware.
- Поліпшені деякі найважливіші методи й області RTL, на базі поліпшень сумісності з популярними бібліотеками C++.
- Поліпшена підтримка Cmake.
- Велика кількість виправлень для підвищення стабільності і якості.
- Відновлення Windows API – Обновлено й додали безліч декларацій API щоб добитися ще більшої інтеграції із платформою Windows.
- Загальні вдосконалення в бібліотеці доступу до БД FireDAC, включаючи оновлені драйвера для FireBird, PostgreSQL і SQLite. Вибір статичного або динамічного підключення SQLite до застосунку.

Змінені стилі VCL для High DPI

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

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

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

Нові High DPI стилі й стилізація окремих VCL компонент

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

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

Поліпшена кроссплатформеність

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

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.

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

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 випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи IP-телефонії з забезпеченням конфіденційності.

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

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

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-122.21.0094.00.00.ПЗ

Арк.

26

Вим.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Опис функцій, що автоматизуються

Структура програмного забезпечення в першу чергу визначається тими функціями, які повинна виконувати програма. В даній роботі автоматизується система керування корпорацією, за рахунок впровадження IP-зв'язку. Для корпоративної мережі зв'язку можливо виділити наступні функції:

- реалізація IP-зв'язку між підрозділами корпорації;
- реалізація селективного зв'язку між підрозділами та структурами;
- реалізація режиму конференції;
- при наявності відповідного устаткування реалізація режиму відеоконференцій;
- реалізація можливості створення бази корпоративних номерів;
- можливість коригування інформації в базі корпоративних номерів.

Розроблюване програмне забезпечення повинне виконувати всі вказані вище функції, але в різному об'ємі. Головне призначення системи організації зв'язку в корпорації, в даному випадку – поліпшити оперативність прийняття рішень, підвищити продуктивність праці, знизити кількість обчислювальних помилок за допомогою автоматизації процесу обробки інформації, сприяти ефективному і безпечному збереженню і доступу до інформації.

Метою корпоративної мережі зв'язку є створення єдиної мережі, що дозволяє ефективно здійснювати керування за рахунок застосування IP-телефонії.

Опис інформаційного забезпечення системи

Під IP-телефонією розуміється технологія, що дозволяє використовувати IP-мережу як засіб організації й ведення міжнародних і міжміських телефонних розмов і передачі факсів у режимі реального часу. Зараз в IP-телефонії існує два

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

основних способи передачі голосових пакетів по IP-мережі: через публічний Інтернет і використовуючи виділені канали. У першому випадку смуга пропускання прямо залежить від завантаженості IP-мережі пакетами, що містять дані, голос, тощо, а значить затримки при проходженні пакетів можуть бути самими різними. При використанні виділених каналів винятково для голосових пакетів можна гарантувати фіксовану (або майже фіксовану) швидкість передачі. Виходячи зі схеми, реалізованої провайдером IP-телефонії, принцип роботи мережі IP-телефонії наступний (розглянутий найпоширеніший випадок – дзвінок з телефону на телефон або з факсимільного апарата на факсимільний апарат).

Необхідно зробити дзвінок на міський телефонний номер телефонного шлюзу, користуючись будь-яким телефонним апаратом або таксофоном, що переходить у тоновий режим. Дзвінок по цифрових або аналогових лініях приходить на телефонний шлюз. Шлюз звертається до сервера голосових повідомлень для видачі голосових підказок і повідомлення залишку на рахунку. Після ідентифікації й автентифікації, пропонується ввести код країни, міста й телефонний номер викликуваного абонента. Все спілкування з телефонним шлюзом відбувається в голосовому каналі. Далі телефонний шлюз встановлює зв'язок з віддаленим телефонним шлюзом по виділеному каналі. Віддалений шлюз робить таке ж з'єднання з викликуваним абонентом, але у зворотному порядку. Після установки з'єднання шлюзи починають обмін IP-Пакетами, у які впакований голос.

Апаратне забезпечення. У якості інтерфейсної плати ISDN використовуються плати для роботи з голосом, які з'єднані послідовно з інтерфейсною платою шиною SCBus. При використанні переривання відлуння одна плата підтримує до 8 ліній, а при використанні знищення ефекту відлуння до 4-х ліній. Зовні шлюз являє собою комп'ютер у промисловому виконанні, що монтується в 19-дюймову стійку. В одній мережі зі шлюзом розташовується голосовий сервер, що відповідає за голосові підказки й повідомлення про залишок на рахунку. Являє собою звичайний IBM-сумісний комп'ютер із ПЗ.

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

Програмне забезпечення. Для кодування використовується стандарт GSM. В IP-мережі голос займає 13,5 Кбіт/сек по протоколу UDP, що також значно зменшує затримки при передачі пакетів. При передачі факсу використовується та ж швидкість, причому використовується протокол гарантованої доставки TCP/IP. Таким чином, при використанні 30 ліній, потрібний канал із шириною пропускання не більше 512 Кбіт/сек, з яких тільки 405 Кбіт/сек будуть задіяні для IP-телефонії. Іншу ширину каналу можна використовувати під Інтернет, причому ніяк не погіршуючи якість передачі голосу. Шлюз підключається до мережі Ethernet, яка підключена до маршрутизатора. Максимально припустима затримка в каналі – 400 мс. При більших затримках недоцільне застосування функцій, що знищують ефект відлуння, тому на програмному рівні відлуння подавлення використовує буферизацію (за замовчуванням в 80 мс).

Рівні архітектури IP-телефонії

Архітектура технології VoIP може бути спрощено представлена у вигляді двох площин. Нижня площина – це базова мережа з маршрутизацією пакетів IP, верхня площина – це відкрита архітектура керування обслуговуванням викликів.

Нижня площина являє собою комбінацію відомих протоколів Інтернет: це – RTP, що функціонує поверх протоколу UDP, розташованого, у свою чергу, у стеці протоколів TCP/IP над протоколом IP. Таким чином, ієрархія RTP/UDP/IP являє собою свого роду транспортний механізм для мовного трафіку. У мережах з маршрутизацією пакетів IP для передачі даних завжди передбачаються механізми повторної передачі пакетів у випадку їхньої втрати. Рекомендації ITU-T допускають затримки в одному напрямку не перевищуючі 150 мс.

Розглянемо протокол TCP/IP. Transmission Control Protocol/Internet Protocol – це промисловий стандарт стеку протоколів, розроблений для глобальних мереж:

– це найбільш завершений стандартний і водночас популярний стек мережних протоколів, що має багаторічну історію.

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

- майже усі великі мережі передають основну частину свого трафіка за допомогою протоколу TCP/IP.
- це метод одержання доступу до мережі Internet.
- стек TCP/IP є основою для створення intranet -корпоративної мережі, що використовує транспортні послуги Internet і гіпертекстову технологію WWW, розроблену в Internet.
- усі сучасні операційні системи підтримують стек TCP/IP.
- це гнучка технологія для з'єднання різноманітних систем як на рівні транспортних підсистем, так і на рівні прикладних сервісів.
- це масштабована міжплатформенна середовище для додатків клієнт-сервер.

Протоколи TCP/IP поділяються на 4 рівні.

Найнижчий IV рівень відповідає фізичному і каналному рівням моделі OSI. Цей рівень у протоколах TCP/IP підтримує всі популярні стандарти фізичного і каналного рівня: для локальних мереж це Ethernet, Token Ring, FDDI, Fast Ethernet, 100VG-AnyLAN, для глобальних мереж – протоколи з'єднань "точка-точка" SLIP і PPP, протоколи територіальних мереж із комутацією пакетів X. 25, frame relay. Звичайно тільки з'являється нова технологія локальних або глобальних мереж вона швидко включається в стек TCP/IP за рахунок розробки відповідного стандарту, що визначає метод інкапсуляції пакетів IP у її кадри.

Наступний III рівень – це рівень міжмережної взаємодії, що займається передачею пакетів із використанням різноманітних транспортних технологій локальних мереж, територіальних мереж, ліній спеціального зв'язку тощо. У якості основного протоколу мережного рівня використовується протокол IP.

Наступний II рівень називається основним. На цьому рівні функціонує протокол керування передачею TCP і протокол дейтаграм користувача UDP. Протокол TCP забезпечує надійну передачу повідомлень між віддаленими прикладними процесами за рахунок утворення віртуальних з'єднань. Протокол UDP забезпечує передачу прикладних пакетів дейтаграмним засобом, як і IP, і

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

виконує тільки функції сполучного ланка між мережним протоколом і численними прикладними процесами.

Верхній I рівень називається прикладним. За довгі роки використання в мережах різноманітних країн і організацій стек TCP/IP накопичив велику кількість протоколів і сервісів прикладного рівня. До них відносять такі широко використовувані протоколи, як протокол копіювання файлів FTP, протокол емуляції терміналу telnet, поштовий протокол SMTP, використовуваний в електронній пошті мережі Internet, гіпертекстові сервіси доступу до віддаленої інформації, такі як WWW і багато інших.

Основу транспортних засобів стека протоколів TCP/IP складає протокол міжмережної взаємодії – Internet Protocol (IP). Основні функції протоколу IP:

- перенос між мережами різноманітних типів адресної інформації в уніфікованій формі,
- складання і розбирання пакетів при передачі їх між мережами з різноманітним максимальним значенням довжини пакета.

Пакет IP складається з заголовка і поля даних. Максимальна довжина поля даних пакета обмежена розрядністю поля, що визначає цей розмір, і складає 65535 байтів, проте при передачі по мережах різноманітного типу довжина пакета вибирається з урахуванням максимальної довжини пакета протоколу нижнього рівня, що несе IP-пакети. Якщо це кадри Ethernet, то вибираються пакети з максимальною довжиною в 1500 байтів, що поміщаються в поле даних кадру.

Задачею протоколу транспортного рівня UDP є передача даних між прикладними процесами без гарантій доставки, тому його пакети можуть бути загублені, продубльовані або прийти не в тому порядку, у якому відправлені.

У стеку протоколів TCP/IP протокол TCP працює так само, як і протокол UDP, на транспортному рівні. Він забезпечує надійне транспортування даних між прикладними процесами шляхом установаження логічного з'єднання.

У протоколі TCP для зв'язку з прикладними процесами використовуються порти. Номера портам присвоюються. Є стандартні, зарезервовані номери

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

(наприклад, номер 21 закріплений за сервісом FTP, 23 – за telnet), а менше відомі додатки користуються довільно обраними локальними номерами.

Перейдемо до верхньої площини керування обслуговуванням запитів зв'язку. Керування обслуговуванням виклику передбачає прийняття рішень про те, куди виклик повинен бути спрямований, і яким образом повинне бути встановлене з'єднання між абонентами. Інструмент такого керування – телефонні системи сигналізації, починаючи із систем, підтримуваних декадно-кроковими АТС і функцій, що передбачають об'єднання, маршрутизації й функцій створення розмовного каналу, що комутирується, у тих самих декадно-крокових шукачах. Далі принципи сигналізації еволюціонували до систем сигналізації по виділених сигнальних каналах, до багаточастотної сигналізації, до протоколів загальканальної сигналізації [6, 7] і до передачі функцій маршрутизації у відповідні вузли обробки послуг мережі [8].

У мережах з комутацією пакетів ситуація більше складна. Мережа з маршрутизацією пакетів IP принципово підтримує одночасно цілий ряд різноманітних протоколів маршрутизації. Такими протоколами на сьогодні є: RIP, IGRP, EIGRP, IS-IS, OSPF, BGP і ін. Точно так само й для IP -телефонії розроблений цілий ряд протоколів

Найпоширенішим є протокол H.323, зокрема, тому, що він став застосовуватися раніше інших протоколів. Інший протокол площини керування обслуговуванням виклику -SIP – орієнтований на те, щоб зробити кінцеві пристрої й шлюзи більш інтелектуальними й підтримувати додаткові послуги для користувачів. Ще один протокол – SGCP – розроблявся, для того, щоб зменшити вартість шлюзів за рахунок реалізації функцій інтелектуальної обробки виклику в централізованому встаткуванні. Протокол IPDC дуже схожий на SGCP, але має більше, ніж SGCP, механізмів експлуатаційного керування (OAM&P). Існує більш функціональний, ніж MGCP, протокол MEGACO. Його адаптований до H.323 варіант в рекомендації H.248.

Щоб стало зрозуміло, чим конкретно відрізняються один від одного

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

перераховані в попередньому параграфі протоколи, розглянемо архітектуру мереж, побудованих на базі цих протоколів, і процедури встановлення й завершення з'єднання з їхнім використанням.

Мережі на базі протоколів H.323 орієнтовані на інтеграцію з телефонними мережами й можуть розглядатися як мережі ISDN, накладені на мережі передачі даних. Рекомендація H.323 передбачає досить складний набір протоколів, що призначений не просто для передачі мовної інформації з IP-мереж з комутацією пакетів. Його мета – забезпечити роботу мультимедійних додатків у мережах з негарантованою якістю обслуговування. Мовний трафік – це тільки один з додатків H.323, поряд з відеоінформацією й даними. Варіант побудови мереж IP-телефонії в рекомендації H.323, добре підходить тим операторам місцевих телефонних мереж, які зацікавлені у використанні мережі з комутацією пакетів (IP-мережі) для надання послуг міжміського й міжнародного зв'язку.

Основними пристроями мережі є: термінал (Terminal), шлюз (Gateway), воротар (Gatekeeper) і пристрій керування конференціями (MCU).

У сценарії встановлення з'єднання між двома користувачами передбачається, що кінцеві користувачі вже знають IP-адреси один одного. У звичайному випадку етапів буває більше, оскільки у встановленні з'єднання беруть участь gatekeeper і й шлюзи.

Розглянемо крок за кроком цей спрощений сценарій.

1. Кінцевий пристрій користувача А надсилає запит з'єднання – повідомлення SETUP – до кінцевого пристрою користувача В.
2. Кінцевий пристрій викликуваного користувача В відповідає на повідомлення SETUP повідомленням ALERTING, що означає, що пристрій вільний, а викликуваному користувачеві подається сигнал про вхідний виклик.
3. Після того, як користувач У приймає виклик, до зухвалої сторони А передається повідомлення CONNECT з номером TCP-порту каналу H.245.
4. Кінцеві пристрої обмінюються по каналі H.245 інформацією про типи використовуваних мовних кодеків та про інші функціональні можливості

						ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			33

встаткування, і сповіщають один одного про номери портів RTP, на які варто передавати інформацію.

5. Відкриваються логічні канали для передачі мовної інформації.

6. Мовна інформація передається в обидва боки в повідомленнях протоколу RTP; крім того, ведеться контроль передачі інформації за допомогою RTCP.

Наведена процедура обслуговування виклику базується на протоколі H.323 версії 1. Версія 2 протоколу H.323 дозволяє передавати інформацію, необхідну для створення логічних каналів, безпосередньо в повідомленні SETUP протоколу H.225.0 без використання протоколу H.245. Така процедура називається «швидкий старт» (Fast Start) і дозволяє скоротити кількість циклів обміну інформацією при встановленні з'єднання. Крім організації базового з'єднання, у мережах H.323 передбачене надання додаткових послуг відповідно до рекомендацій ITU H.450.X. Моніторинг якості обслуговування забезпечується протоколом RTCP, однак обмін інформацією RTCP відбувається тільки між кінцевими пристроями, що беруть участь у з'єднанні.

Другий підхід до побудови мереж IP-телефонії заснований на використанні протоколу SIP – Session Initiation Protocol. SIP являє собою текст – орієнтований протокол, що є частиною глобальної архітектури мультимедіа. Ця архітектура також містить у собі протокол резервування ресурсів (RSVP, RFC 2205), транспортний протокол реального часу (RTP, RFC 1889), протокол передачі потоків у реальному часі (RTSP, RFC 2326), протокол опису параметрів зв'язку (SDP, RFC 2327), протокол повідомлення про зв'язок (SAP). Однак функції протоколу SIP не залежать від кожного із цих протоколів.

Третій підхід до побудови мереж IP-телефонії, заснована на використанні протоколу MGCP. При розробці цього протоколу робоча група MEGACO опиралася на мережну архітектуру, що містить основні функціональні блоки трьох видів:

– шлюз – Media Gateway (MG), що виконує функції перетворення мовної інформації, що надходить із боку ТфОП з постійною швидкістю передачі, у ви д,

придатний для передачі по мережах з маршрутизацією пакетів IP (кодування й упакування мовної інформації в пакети RTP/UDP/IP, та зворотнє перетворення);

- контролер шлюзів – Call Agent, що виконує функції керування шлюзами;
- шлюз сигналізації – SIGNALING Gateway (SG), що забезпечує доставку сигнальної інформації, що надходить із боку ТфОП, до контролера шлюзів і перенос сигнальної інформації у зворотному напрямку.

Таким чином, весь інтелект функціонально розподіленого шлюзу зосереджений у контролері, функції якого можуть бути розподілені між декількома комп'ютерними платформами.

Для побудови добре функціонуючих і сумісних із ТфОП мереж IP-телефонії підходять протоколи H.323 і MGCP. Як вже відзначалось, протокол SIP трохи гірше взаємодіє із системами сигналізації, використовуваними в ТфОП. Підхід, заснований на використанні протоколу MGCP, має досить важливу перевагу перед підходом H.323: підтримка контролером шлюзів сигналізації ОКС7 і інших видів сигналізації, а також прозора трансляція сигнальної інформації з мережі IP-телефонії. У мережі, побудованої на базі рекомендації H.323, сигналізація ОКС7, як і будь-яка інша сигналізація, конвертується шлюзом у сигнальні повідомлення H.225.0 (Q.931). Основним недоліком третього з наведених у даному параграфі підходів є незакінченість стандартів. До недоліків можна віднести також відсутність стандартизованого протоколу взаємодії між контролерами. Крім того, протокол MGCP є протоколом керування шлюзами, але не призначений для керування з'єднаннями за участю термінального устаткування користувачів (IP-телефонів). Це означає, що в мережі, побудованої на базі протоколу MGCP, для керування термінальним устаткуванням повинен бути присутнім gatekeeper або сервер SIP. Варто також відзначити, що в існуючих додатках IP-телефонії, таких як надання послуг міжнародного й міжміського зв'язку, використовувати протокол MGCP (також, як і протокол SIP) недоцільно у зв'язку з тим, що основна кількість мереж IP-телефонії сьогодні побудована на базі протоколу H.323. В останньому зі згаданих підходів (у проекті

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

версії 4 рекомендації H.323) введено принцип декомпозиції шлюзів, використаний у третьому підході. Керування функціональними блоками розподіленого шлюзу буде здійснюватися контролером шлюзу – MGC (Media Gateway Controller) за допомогою протоколу MEGACO/H.248.

Опис протоколу обміну інформацією

Як було обґрунтовано раніше, сучасна IP-телефонія будується на основі протоколу H.323. Розглянемо більш докладно цей протокол. У рекомендаціях, що входять у сімейство H.323, визначені протоколи, методи й мережні елементи, необхідні для організації мультимедійного зв'язку між двома або більше користувачами [15-22].

Найбільш затребуваною з послуг, специфікованих у рекомендації H.323, є послуга передачі мовної інформації з мереж з маршрутизацією пакетів IP. Найпоширенішим підходом до побудови мереж IP-телефонії сьогодні є саме підхід, запропонований ITU-T у рекомендації H.323.

Сімейство протоколів H.323 містить у собі три основних протоколи: протокол взаємодії кінцевого устаткування з gatekeeper – RAS, протокол керування з'єднаннями – H.225 і протокол керування логічними каналами – H.245. Ці три протоколи, разом з Інтернет-протоколами TCP/IP, UDP, RTP і RTCP, а також з описаним в [6] протоколом Q.931 складають основу технології IP-телефонії. Суть ієрархії цих протоколів полягає в наступному. Для переносу сигнальних повідомлень H.225 і керуючих повідомлень H.245 використовується протокол з встановленням з'єднання й з гарантованою доставкою інформації – TCP. Сигнальні повідомлення RAS переносяться протоколом з негарантованою доставкою інформації – UDP. Протокол RAS забезпечує контроль використання мережних ресурсів, підтримує автентифікацію користувачів і може забезпечувати нарахування плати за послуги. Для переносу мовної і відеоінформації використовується протокол передачі інформації в реальному часі – RTP. Контроль переносу користувальницької інформації виробляється протоколом RTCP. Процедура встановлення з'єднання в таких мережах IP-телефонії базується

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

на рекомендації Q.931 [15] і аналогічна процедурі, використовуваної в ISDN.

Основними пристроями мережі є: термінал, шлюз, gatekeeper і пристрій керування конференціями.

Термінал H.323 – це кінцевий пристрій мережі IP-телефонії, що забезпечує двосторонній мовний або мультимедійний зв'язок з іншим терміналом, шлюзом або пристроєм керування конференціями. Користувальницький інтерфейс керування системою дає користувачеві можливість створювати й приймати виклики, а також конфігурувати систему й контролювати її роботу.

Таблиця. 3.1 – Сімейство протоколів H.323

Гарантована доставка інформації із протоколу TCP		Негарантована доставка інформації із протоколу UDP		
H.245	H.225	Потоки мови й відеоінформації		
	Керування з'єднанням (Q.931)	RAS	RTCP	RTP
TCP		UDP		
IP				
Канальний рівень				
Фізичний рівень				

Модуль керування підтримує три види сигналізації: H.225, H.245 і RAS. Цей модуль забезпечує реєстрацію терміналу у gatekeeper, установлення й завершення з'єднання, обмін інформацією, необхідної для відкриття мовних каналів, надання додаткових послуг і техобслуговування.

Телематичні додатки забезпечують передачу користувальницьких даних, нерухоливих зображень і файлів, доступ до баз даних і т.п. Стандартним протоколом для підтримки таких додатків є протокол T. 120.

Модуль H.225.0 відповідає за перетворення відеоінформації, мови, даних і сигнальної інформації у вид, придатний для передачі по мережах з

маршрутизацією пакетів IP, і за зворотне перетворення. Крім того, функціями модуля є розбивка інформації на логічні кадри, нумерація послідовно переданих кадрів, виявлення й корекція помилок.

Мережний інтерфейс забезпечує гарантовану передачу керуючих повідомлень H.245, сигнальних повідомлень H.225.0 (Q.931) і користувальницьких даних за допомогою протоколу TCP і негарантовану передачу мовної й відеоінформації, а також повідомлень RAS, за допомогою протоколу UDP.

Блок синхронізації вносить затримку на прийомній стороні з метою забезпечити синхронізацію джерела інформації з її приймачем, узгодження мовних і відеоканалів або згладжування варіації затримки інформації.

Відеокодеки кодують відеоінформацію, що надходить від зовнішнього джерела відеосигналів (відеокамери або відеомагнітофона), для її передачі по мережі з маршрутизацією пакетів IP і декодують сигнали, що надходять із мережі, для наступного відображення відеоінформації на моніторі або телевізорі.

Аудіокодеки кодують аудіоінформацію, що надходить від мікрофона (або інших джерел аудіоінформації), для її передачі по мережі з маршрутизацією пакетів IP і декодують сигнали, що надходять із мережі, для відтворення.

Слід зазначити, що при організації децентралізованої конференції термінал H.323 може приймати більш ніж один потік мовної інформації. У цьому випадку термінал повинен уміти змішувати або перемикати пакетовану мову, що надходить від інших учасників конференції.

Основною функцією шлюзу H.323 є перетворення мовної (мультимедійної) інформації, що надходить із боку ТфОП з постійною швидкістю, у вид, придатний для передачі по IP-мережах.

При відсутності в мережі gatekeeper повинна бути реалізована ще одна функція шлюзу – перетворення номера ТфОП у транспортну адресу IP-Мережі.

З боку мереж з маршрутизацією пакетів IP, так само, як і з боку ТфОП, шлюз може брати участь у з'єднаннях як термінал або пристрій керування

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

конференціями. У випадку, коли термінал H.323 зв'язується з іншим терміналом H.323, розташованим у ті ж самій IP-мережі, шлюз у цьому з'єднанні не бере участь.

Шлюз, у сукупності з gatekeeper мережі IP-телефонії, утворить універсальну платформу для надання всього спектра послуг зв'язку.

У gatekeeper зосереджений весь інтелект мереж IP-телефонії, що базуються на рекомендації ITU H.323. Мережа H.323 має зонну архітектуру. Gatekeeper виконує функції керування зоною мережі IP-телефонії, у яку входять термінали, шлюзи й пристрої керування конференціями, зареєстровані в цього gatekeeper. Різні ділянки зони мережі H.323 можуть бути територіально рознесені й з'єднуватися один з одним через маршрутизатори. Варто звернути увагу на те, що комутатори кадрів Ethernet і маршрутизатори пакетів IP не є мережними елементами H.323, тому що вони працюють на ланковому або мережному рівнях відповідно, у той час як устаткування H.323 працює на прикладному рівні стека протоколів TCP/IP.

У число найбільш важливих функцій, виконуваних gatekeeper, входять:

- перетворення так званої alias-адреси (ім'я абонента, телефонного номера, адреси електронної пошти й ін.) у транспортну адресу мережі з маршрутизацією пакетів IP (IP адреса й номер порту TCP);
- контроль доступу користувачів системи до послуг IP-телефонії за допомогою сигналізації RAS (використовуються повідомлення ARQ/ACF/ARJ);
- контроль, керування й резервування пропускної здатності мережі;
- маршрутизація сигнальних повідомлень між терміналами, розташованими в одній зоні; gatekeeper може організовувати сигнальний канал безпосередньо між терміналами або ретранслювати сигнальні повідомлення від одного терміналу до іншого.

У тому випадку, коли абонент, який викликає, знає IP-адресу терміналу абонента, який викликається, з'єднання між двома пристроями може бути встановлене без допомоги gatekeeper. Але, при наявності gatekeeper я

											Арк.
											39
Вим.	Арк.	№ докум.	Підпис	Дата	ВКРМ-122.21.0094.00.00.ПЗ						

забезпечується мобільність абонентів, тобто здатність користувача одержати доступ до послуг з будь-якого терміналу в будь-якому місці мережі й здатність мережі ідентифікувати користувачів при їхньому переміщенні з одного місця в інше. При відсутності в мережі gatekeeper перетворення адреси викликуваного абонента, що надходить із боку ТфОП у форматі E. 164, у транспортну адресу IP-мережі повинне виконуватися шлюзом. В одній мережі може перебувати декілька gatekeeper, які повинні взаємодіяти між собою. Слід особливо зазначити, що хоча gatekeeper є окремим логічним елементом мережі, він може бути реалізований у терміналі, у шлюзі, у пристрої керування конференціями або в пристроях, не специфікованих у рекомендації Н.323.

Рекомендація Н.323 передбачає три види конференцій.

Перший вид – централізована конференція, у якій кінцеві пристрої з'єднуються в режимі точка-точка із пристроєм керування конференціями (MCU), що контролює процес створення й завершення конференції, а також обробку потоків користувацької інформації.

Другий вид – децентралізована конференція, у якій кожний її учасник з'єднується з іншими учасниками в режимі точка – група точок, і кінцеві пристрої самі обробляють потоки інформації, що надходять від інших учасників.

Третій вид – змішана конференція, тобто комбінація двох попередніх видів.

Перевага централізованої конференції – порівняно прості вимоги до термінального встаткування, недолік – більша вартість пристрою керування конференціями.

Для децентралізованої конференції потрібно більше складне термінальне встаткування, крім того, бажано, щоб у мережі підтримувалася передача пакетів IP у режимі багатоадресного розсилання (IP multicasting). Якщо мережа не підтримує цей режим, термінал може передавати інформацію до кожного з інших терміналів, що беруть участь у конференції, у режимі точка-точка, але це стає неефективним при числі учасників більше чотирьох. Пристрій керування конференціями MCU містить один обов'язковий елемент – контролер

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

багатоточкових з'єднань – Multipoint controller (MC). Крім того, MCU може містити один або більше процесорів для обробки інформації користувачів при багатоточкових з'єднаннях – Multipoint processor (MP). Слід зазначити, що контролер MC і процесор MP є самостійними логічними пристроями H.323 і що контролер може існувати незалежно від процесора (зворотне невірно). Контролер може бути фізично сполучений з gatekeeper, зі шлюзом або з MCU, а MCU, у свою чергу, може бути сполучене зі шлюзом або з gatekeeper.

Контролер конференцій повинен використовуватися для організації конференції будь-якого виду. Він організує обмін між учасниками конференції даними про функціональні можливості їхніх терміналів, указує, у якому режимі (з використанням яких кодеків) учасники конференції можуть передавати інформацію, причому цей режим може змінюватися в ході конференції, наприклад, при підключенні до неї нового учасника. Таким чином, контролер MC визначає режим конференції, що може бути загальним для всіх учасників конференції або окремим для кожного з них. В зв'язку з тим, що в мережі може бути кілька контролерів MC, то для кожної знову створюваної конференції повинна проводитися процедура визначення встаткування, для того, щоб виявити той з контролерів MC, що буде управляти даною конференцією. При організації централізованої конференції, крім контролера MC, повинен використовуватися процесор MP, що обробляє користувальницьку інформацію й відповідає за перемикання або змішування мовних потоків, відеоінформації й даних. При організації децентралізованої конференції процесор MP не використовується.

3.2 Розробка структурної схеми

Для перевірки коректності реалізації даної задачі було виконано багато розрахунків та експериментальних матеріалів. Цьому питанню приділялась особлива увага тому, що помилка при розрахунку привела б до ряду негативних

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

наслідків. Відлагодження та перевірка, що підтверджує вірність програмних рішень відбувалась за декількома етапами:

- математична перевірка окремих модулів;
- математична перевірка всієї системи (з допомогою математичної логіки будується логічна схема всієї системи);
- практична перевірка підпрограм (перевіряється процедурна частина кожної підпрограми окремо);
- практична перевірка всієї системи у дії (перевіряється система в цілому за допомогою вводу різних даних у програму, потім на виході з програми перевіряємо отриману інформацію з очікуваною).

Для підтвердження правильності розрахунку програми були використані експериментальні дані різних аудіо форматів, були проведені консультації з даного питання зі спеціалістами.

Простота мови проектування та маніпулювання даними, зручність спілкування користувача з системою до мінімуму вивчення цієї програми. Користувач програми – це людина, яка повинна володіти азами програмування. При написанні програми я намагалася, щоб програма відповідала наступним параметрам:

- Швидкодія. Програма працює постійно з великою кількістю кінцевих абонентів (селективний зв'язок).
- Захист. Забезпечити надійний захищений канал зв'язку.
- Відсутність проблеми дороговизни сучасних персональних комп'ютерів. Система, що написана може встановлюватись на будь-якому персональному комп'ютері – використовувати відносно швидкі алгоритми захисту зв'язку.
- Можливість зручно і швидко формувати приклади і теорію для користувача.
- Можливість звертання до системних ресурсів. Користувача системи цікавить її інформаційний та сенсовий зміст. Подробиці організації фізичного

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

зберігання даних його не цікавлять.

Перш за все перед розробкою системи слід одержати уявлення на наступні моменти:

- на які частини можна розбити систему;
- одержати уявлення про кожну частину (фрагмент);
- яка інформація і з якою детальністю необхідна користувачу кожного фрагменту;
- які процеси передачі і обробки даних знаходяться в кожному фрагменті;
- технологія накопичування і обробки аудіо інформації системи;
- на якому обладнанні планується реалізувати систему;
- технологія функціонування системи;
- чи необхідна адаптація і налаштування системи при змінах деяких умов.

Для розробки програми були попередньо збудовані функціональна схема, структурна схеми, структурна схема системи керування, схема процесів, а також блок-схеми алгоритму програми, розглянемо їх детально.

В процесі практичної реалізації теоретичних принципів розробки системи, додатково розглянутих вище, була розроблена структурна схема системи, яка зображена на рисунку 3.1.

Завдяки структурній схемі можна чітко побачити основні структурні блоки системи та взаємозв'язки між ними. При розробці структурної схеми основний упор робився на існуючі розробки ПЗ і їх модулі допомоги. Аналіз рисунка 3.1 дозволяє чітко прослідити як працює програма.

Розглянемо схему зверху вниз, в напрямку від пристрою до кінцевої програми – за допомогою внутрішнього пула доступу що забезпечує закриті канали зв'язку абоненти генеральний директор, відділи бухгалтерії, логістики, складу, збуту, економічної безпеки – можуть взаємодіяти з дочірніми виробниками з використанням внутрішньої АТМ. і чітко налагодженої технічної системи взаємодії. Технічна частина взаємодії забезпечується VoIP та вогоратарем з використанням маршрутизатора.

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

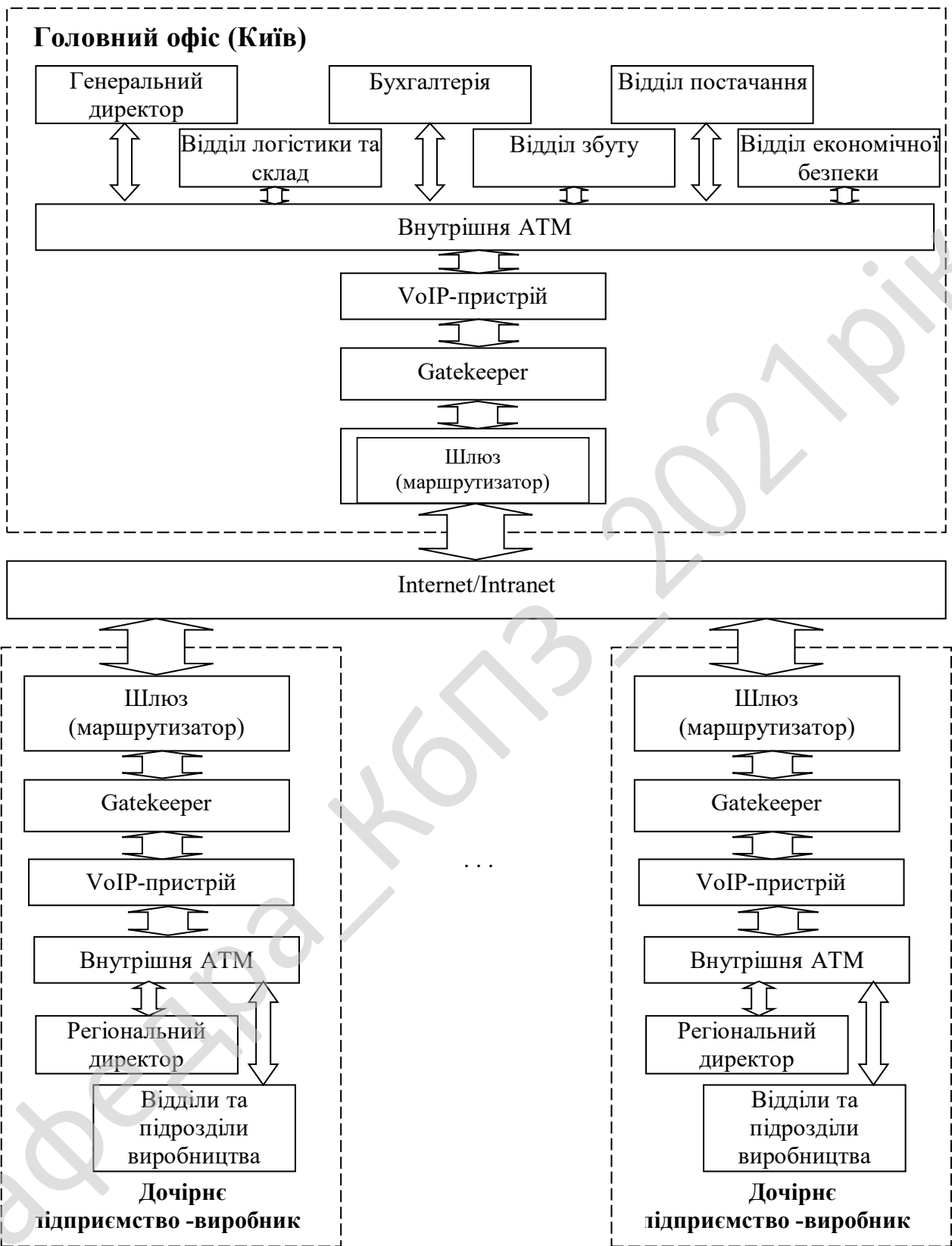


Рисунок 3.1 – Структурна схема системи

3.3 Розробка функціональної схеми

На функціональній схемі, зображеній на рисунку 3.2, є можливість прослідкувати за можливими шляхами проходження сигналів від одного функціонального блоку до іншого.

Тонка стрілочка вказує шлях проходження сигналів між функціональними блоками, товста – мережна взаємодія.

Розглянемо основні функціональні взаємодії між головним офісом та дочірніми підприємствами. З головного офісу з застосуванням ІР технології проводиться зв'язок з дочірніми підприємствами, такими абонентами як – регіональний директор, підрозділи.

Є можливість проведення різних режимів зв'язку при різній кількості абонентів з обліком абонентів (за допомогою БД).

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

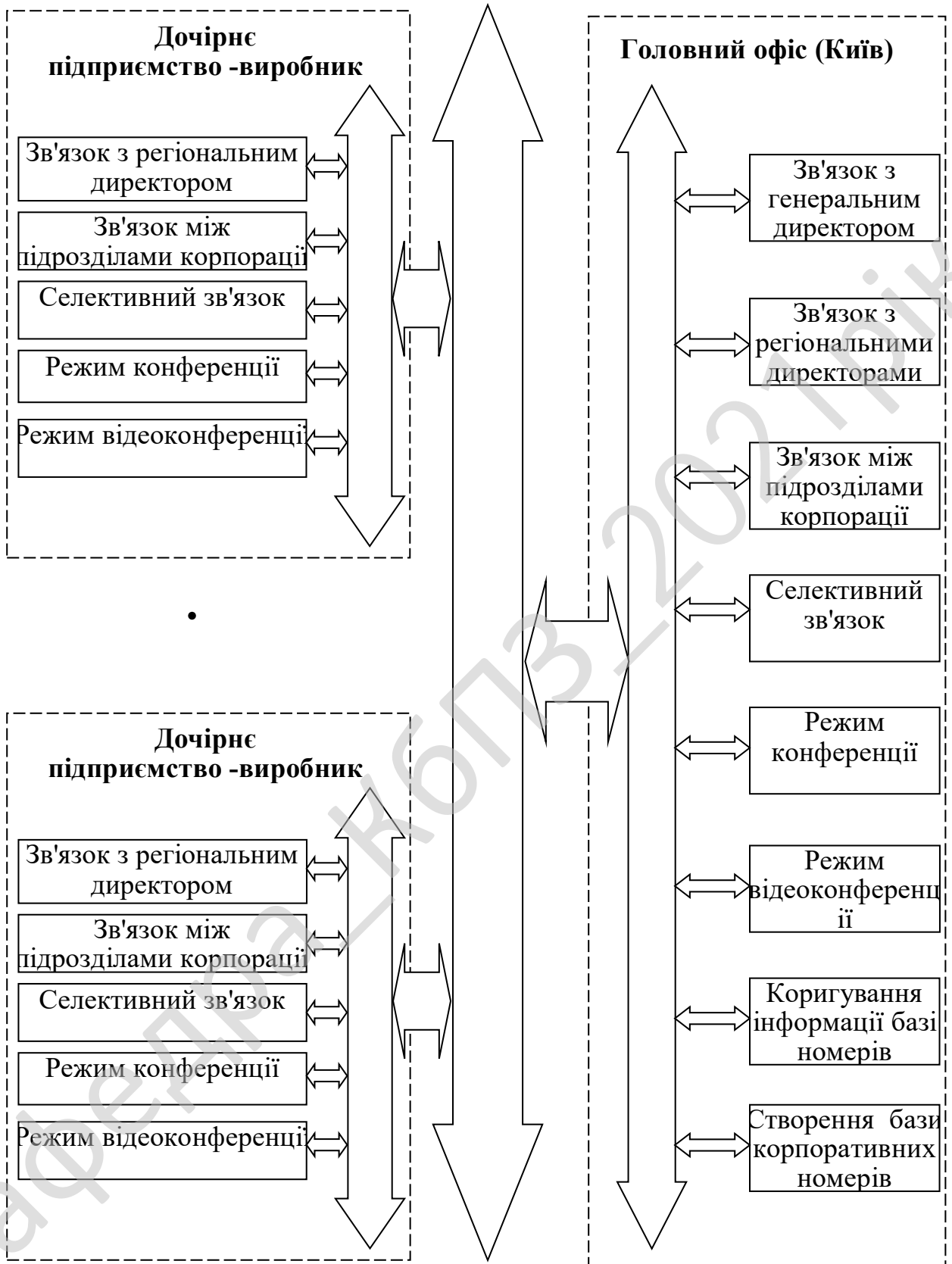


Рисунок 3.2 – Функціональна схема системи

3.4 Розробка діаграми процесів

Одним з важливих критеріїв при розробці будь-якого програмного забезпечення це грамотна розробка структури роботи системи потоків і процесів (рисунок 3.3). Розглянемо діаграму процесів. На діаграмі процесів можна точно зрозуміти як працює і взаємодіє ПЗ в цілому.

Починається і закінчується програма в першому блоці є основною крапкою відрахунку діаграми – «Основний блок ПЗ (початок/кінець)». При переміщенні по стрілках можна побачити загальну схему взаємодії блоків і їх входження один в одного.

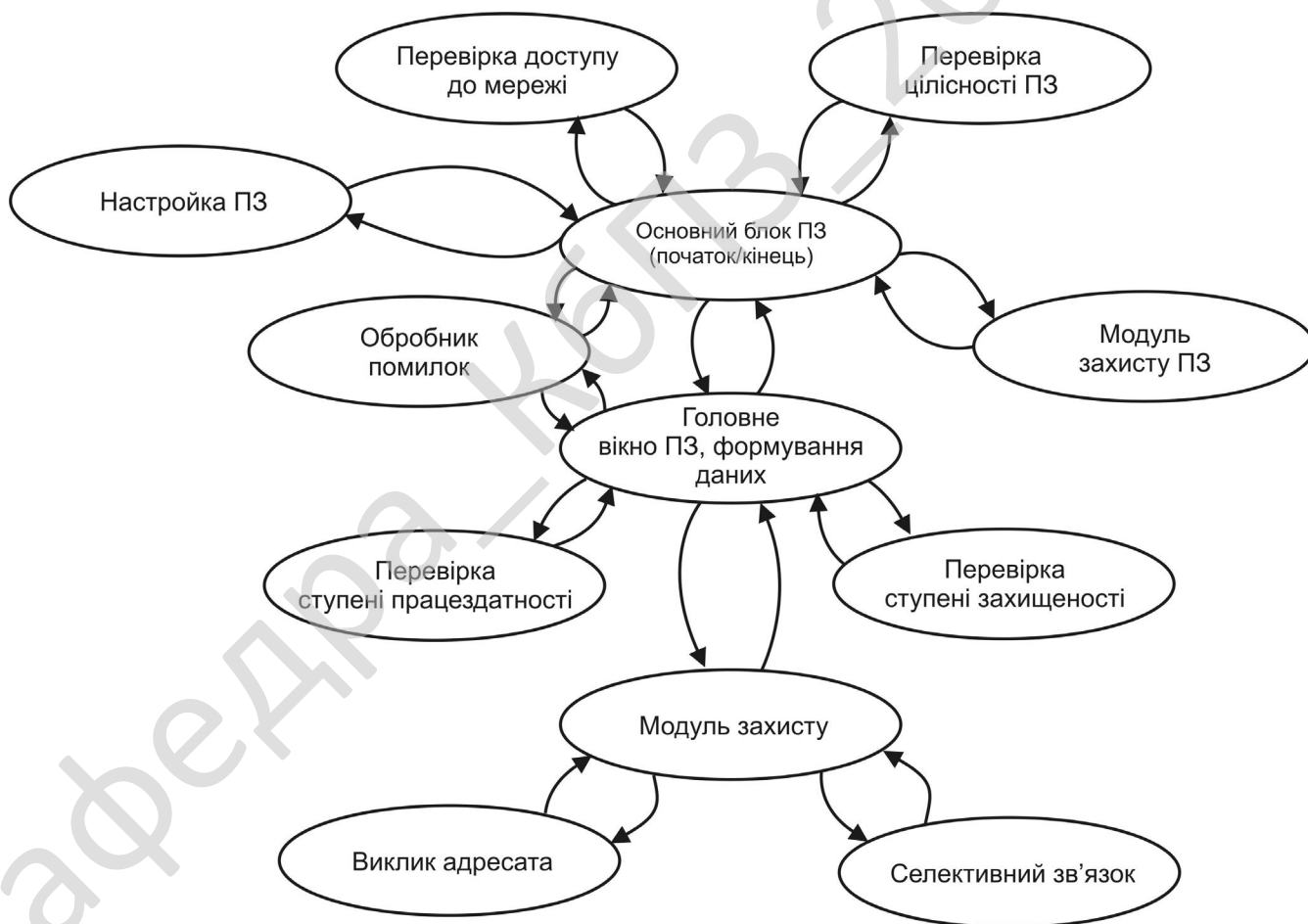


Рисунок 3.3 – Схема взаємодії процесів

Чітко простежується взаємодія основного блоку програми з первинними функціями програми без яких надійне функціонування ПЗ неможливе. Перед тим як зв'язатися з кінцевим абонентом ПЗ перевіряє ступень захищеності та ступень працездатності каналу зв'язку, проходить модуль захисту ПЗ та Викликає кінцевого адресата чи групу адресатів (селективний зв'язок). При виникненні помилки ПЗ запускає обробник помилок і локалізує помилку.

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

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

Первинною стадією без якої не відбувається розробка програмного забезпечення це звичайно розробка блок-схем. Блок-схеми є першоджерелами стратегії розвитку ПЗ. Тому від точності і детальної блок-схеми залежить результат всієї програми. При виборі початкової точки відліку при побудові схем я враховував, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об'єкно-модульна, проект, що розробляється, вимагає оптимізації програми високого рівня, також те, що при розробці програми слід надати особливу увагу модулям захисту каналу зв'язку з використанням IP технології, модулю обробки помилок програми і аудіо модулю. Створення повноцінного коду програмного продукту з розробленою ціною політикою і системою знижок покупцям програмної продукції.

На рисунку 4.1 зображена основна блок-схема програми. При детальному розгляді програма розбита на декілька основних важливих блоків чи групи блоків, таких як:

- Блок ініціалізації початкових значень типів даних, констант, змінних, класів, масивів і графічних даних.
- Блок підключення додаткових файлів.
- Блок пошуку та читання файлів програмного забезпечення, якщо файли ПЗ не знайдені відбуваються дії для завершення ПЗ з виведенням повідомлення. Локальне повідомлення реалізоване WIN API32 функцією MessageDlg.
- Блоків налагодження на робочий режим.
- Блок підпрограми зв'язку.
- Блоків завершення роботи програми.

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

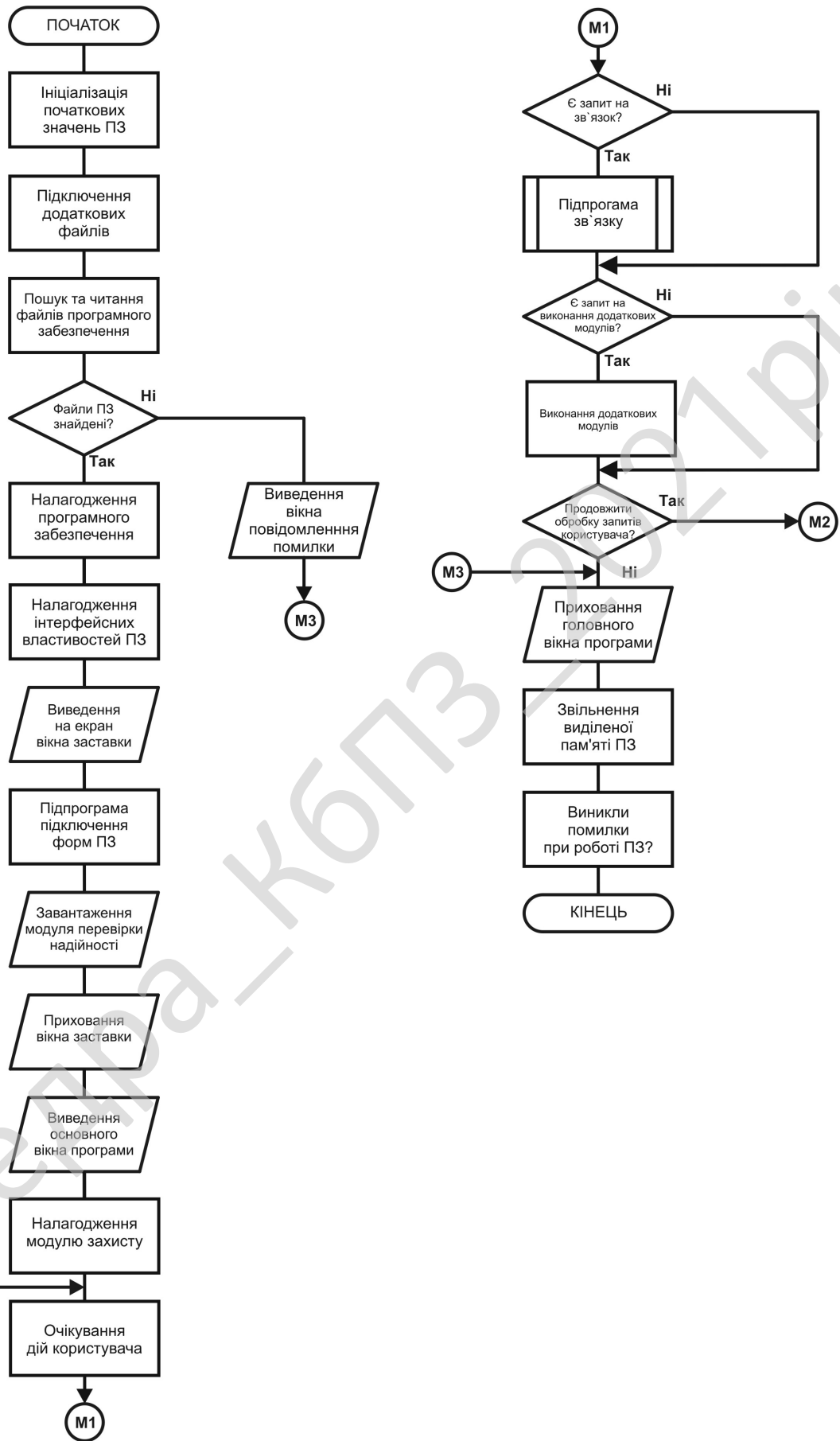


Рисунок 4.1 – Блок-схема програми

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-122.21.0094.00.00.ПЗ

Арк.

50

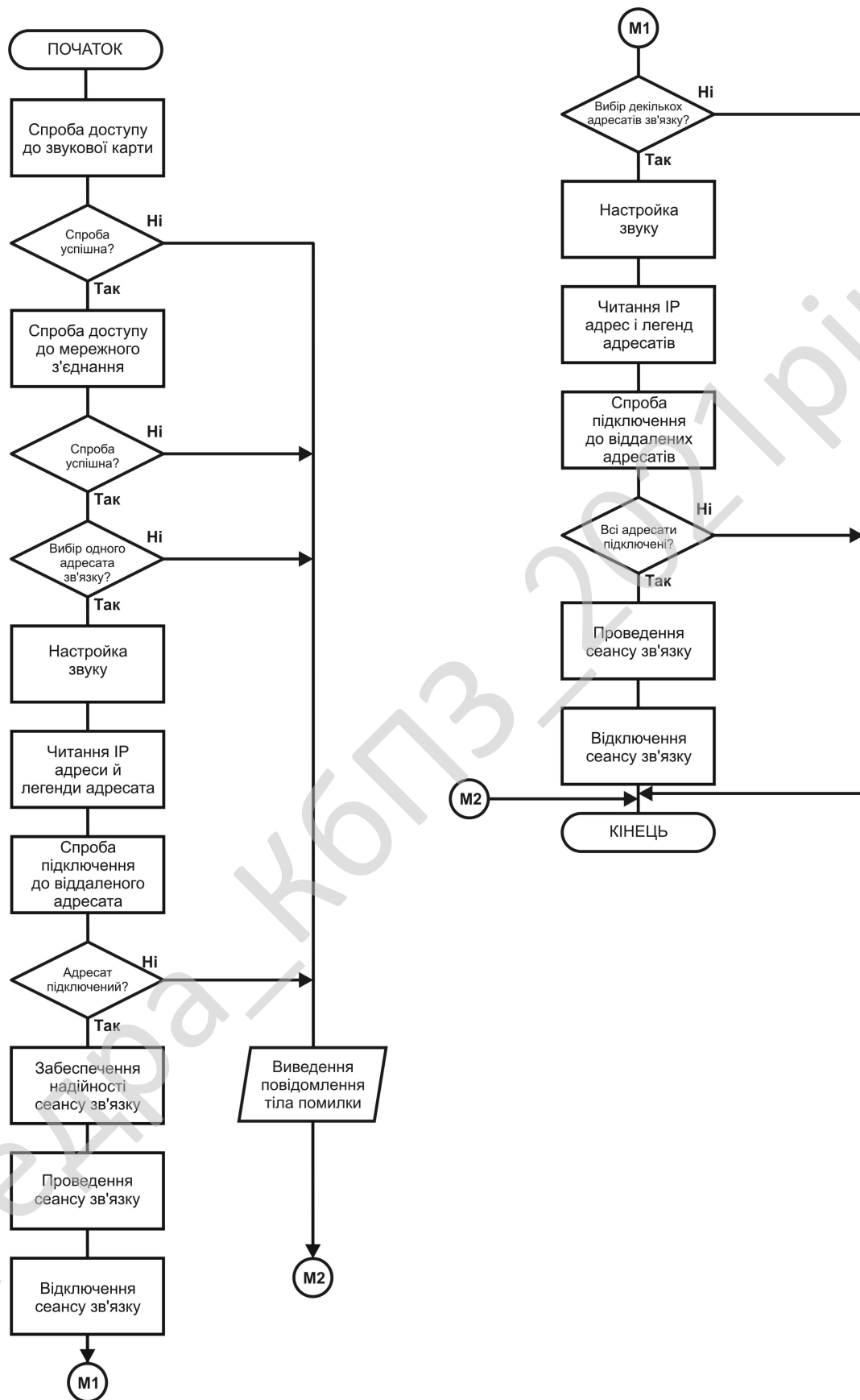


Рисунок 4.2 – Блок-схема підпрограми зв'язку

Блок підпрограми зв'язку відображений на рисунку 4.2 і описує основний алгоритм аналізу та роботи каналу зв'язку.

Саме в цьому блоці в залежності від вибору адресата проводиться спроба доступу до мережного з'єднання (internet/intranet), вибір адресата чи адресатів зв'язку, налаштування та проведення сеансу зв'язку.

Опис математичного забезпечення

Для передачі мови по IP-каналам необхідно, як було відмічено раніше, перетворити аналоговий сигнал у цифровий, тобто безперервний в дискретний. Виявляється, що навіть такі на перший погляд зовсім різні сигнали, як безперервні й дискретизовані мають дуже багато загального й зв'язані твердою функціональною залежністю, встановленою теоремою дискретизації, або теоремою Котельникова [3].

Реальний безперервний сигнал володіє спектром, основна частина енергії якого зосереджена в обмеженій смузі частот. Це зумовлене тим, що прилади, що формують і перетворюють повідомлення і сигнали, а також канали зв'язку мають кінцеву смугу пропускання. Функція часу з обмеженням по ширині спектром повністю визначається своїми миттєвими значеннями, відрахованими через інтервали часу:

$$\Delta t = 1/2F_m, \quad (4.1)$$

де F_m - найвища частота спектру сигналу.

Це положення складає зміст теореми Котельникова

Теорема дискретизації, або, як її ще називають, теорема Котельникова, теорема Уїтекера, формулюється в такий спосіб: безперервна функція $X(t)$ з обмеженим спектром, тобто не має у своєму спектрі

$$F\{X(t)\} = \int_{-\infty}^{\infty} X(t) \cdot e^{-j2\pi ft} dt \quad (4.2)$$

складових із частотами, що лежать за межами смуги $f \in (-F_m, F_m)$, повністю визначається послідовністю своїх обчислень у дискретні моменти часу $X(t_i)$, що виходять із кроком $\Delta t < 1/F_m$. Іншими словами безперервна

детермінована функція часу $X(t)$, що має обмежений спектр, може бути розкладена в ряд по ортогональним функціям часу виду:

$$\Psi_i(t) = \frac{\sin 2F_m \pi (t - i\Delta t)}{2\pi F_m (t - i\Delta t)} \quad (4.3)$$

з коефіцієнтами, рівними значенням функції $X(i\Delta t)$. Цей розклад, що називається рядом Котельникова, має наступний вигляд: $X(t) = \sum_{i=-\infty}^{\infty} X(i\Delta t)\Psi_i(t)$.

Функція відрахунків має наступні властивості:

1. Вона дорівнює 1 при $t = i\Delta t$ і рівна 0 в усіх інших точках $t = j\Delta t; j \neq i$
2. Її спектр рівномірний в межах смуги частот від $-F_{\max}$ до F_{\max} але рівний 0 поза цією смугою.

3. Вона має властивості ортогональності:

$$\int_{-\infty}^{\infty} \Psi_i(t)\Psi_j(t) dt = \begin{cases} 0 & \text{при } i \neq j \\ \frac{1}{2F_{\max}} & \text{при } i = j \end{cases}$$

Якщо функція $f(t)$ з обмеженим спектром розглядається на кінцевому інтервалі часу T , то число дискрет, що передаються, буде дорівнювати: $T/\Delta t = 2F_m T$. Величину $F_m T$ називають базою сигналу.

Доказ сформульованої теореми ґрунтується на однозначній відповідності між сигналами й відповідними їм спектрами. Іншими словами, якщо сигнали однакові, те й відповідні їм спектри також однакові. І, навпаки, якщо спектри двох сигналів однакові, той і відповідний сигнали також однакові.

Опис технологічного процесу обробки даних

Технологічний процес складається з двох основних етапів – збір і облік даних по IP-адресам й формування відповідної легенди та власне сама робота програми у режимі IP-телефонії та селективного зв'язку. Вони можуть виконуватися в будь-який календарний момент часу і включають операції введення, з'єднання і ін. Операції мають програмне виконання, підлегле єдиній алгоритмічній схемі. Програма реалізована в середовищі Delphi. Робота з

				ВКРМ-122.21.0094.00.00.ПЗ		Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		
						53

програмою починається з виведення інформаційного вікна й активізації системи меню й здійснюється в діалоговому і по-дійному режиму. При цьому під діалогом розуміється надання користувачу декількох альтернатив і обробка його вибору. У діалогову систему входять головне меню з відповідними спливаючими підменю а також діалогові вікна. Під подіями розуміються процеси, що активізуються користувачем (наприклад – натискання функціональних клавіш), а також програмні події – з'єднання по IP-технології.

Програма складається з наступних основних модулів.

Основна процедура – конфігурація середовища оточення, формування основного екрана програми, створення системи головного меню і відповідних підменю, активізація меню.

Процедура обробки головного меню – запуск відповідної процедури. Процедура введення даних – забезпечення введення інформації у бази даних IP-адрес та легенд до них, контроль за допустимістю значень, забезпечення введення даних шляхом вибору зі списку.

Допоміжні процедури і функції – реалізація запитів, повідомлень, формування списків вибору IP-адрес та легенд, а також контроль за даними, що вводяться.

Усі модулі в програмі зв'язані між собою за даними, що аналізуються на вході і виробляються на виході. Дані в модулі надходять через діалог з користувачем, параметри і документи інформаційної бази. Передача даних від одного модуля до іншого здійснюється тільки через збережені документи.

Дані через діалог можуть бути отримані прямим і непрямим способом. Прямий спосіб реалізується шляхом їхнього введення за шаблоном чи по запиті конкретних значень реквізитів. Непрямий спосіб – шляхом чи меню логічних (альтернативних) запитів – «так», «ні». При непрямому способі дані, що надходять у модуль, заздалегідь передбачені алгоритмом, але зовні виглядають в обліку відомими фразами.

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

Параметри (мова) – вхідні дані, отримані у виді конкретних значень, переданих в оперативній пам'яті суміжним модулям (функціям).

Обґрунтування інформаційного забезпечення

Перелік первісних даних

Під вхідною інформацією розуміється вся інформація, необхідна для вирішення задачі і розташована на різних носіях: первинних документах, машинних носіях, у пам'яті персонального комп'ютера. Вхідною інформацією для розроблювальної в дипломному проєкті корпоративної системи зв'язку з використанням IP-технологій є мова або відеозображення.

Щоб передати мову через телефонну мережу, мовну інформацію потрібно перетворити в аналоговий електричний сигнал. При переході до цифрових мереж зв'язку виникла необхідність перетворити аналоговий електричний сигнал у цифровий формат на передавальній стороні, тобто закодувати, і перевести назад в аналогову форму, тобто декодувати, на прийомній стороні [5-8].

Процес перетворення аналогового мовного сигналу в цифрову форму називають аналізом або цифровим кодуванням мови, а зворотний процес відновлення аналогової форми мовного сигналу – синтезом або декодуванням мови.

Ціль будь-якої схеми кодування – одержати таку цифрову послідовність, що вимагає мінімальної швидкості передачі й з якої декодер може відновити вихідний мовний сигнал з мінімальними змінами.

При перетворенні мовного сигналу в цифрову форму, мають місце два процеси – дискретизація, тобто формування дискретних у часі відрахунків амплітуди сигналу, і квантування, тобто дискретизація отриманих значень за амплітудою. Ці дві функції виконуються т.зв. аналого -цифровими перетворювачами (АЦП), які розміщуються в сучасних АТС на платі абонентських комплектів, а у випадку передачі мови по IP -мережах – у терміналі користувача (комп'ютері або IP-телефоні).

Процес аналого-цифрового перетворення одержав, стосовно до систем

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

зв'язку, назву імпульсно-кової модуляції (ІКМ).

Щоб знизити необхідну швидкість передачі біт, застосовують нелінійний (логіфмічний) закон квантування, тобто квантуванню піддається не амплітуда сигналу, а її логарифм. У цьому випадку має місце процес «стиску» динамічного діапазону сигналу, а при відновленні сигналу відбувається зворотний процес.

Сьогодні застосовуються два основні різновиди ІКМ: з кодуванням по m -закону й по А-закону. У результаті стиску сигнал з амплітудою, що кодується 12 - 13 бітами, описується всього вісьма бітами. Розрізняються ці різновиди ІКМ деталями процесу стиску (m -закон кодування переважніше використовувати при малій амплітуді сигналу й при малому відношенні сигнал/шум). У Північній Америці використовується кодування по m -закону, а в Європі – по А-Закону. Тому при міжнародному зв'язку в багатьох випадках потрібне перетворення m -закону в А-закон, відповідальність за яке несе країна, у якій використовується m -закон кодування. В обох випадках кожний відлік кодується 8 бітами, або одним байтом, який можна вважати звуковим фрагментом. Для передачі послідовності таких фрагментів необхідна пропускна здатність каналу, рівна 64 Кбіт/с. Оскільки ІКМ була першою стандартною технологією, що одержала широке застосування в цифрових системах передачі, пропускна здатність каналу, рівна 64 Кбіт/с, стала всесвітнім стандартом для цифрових мереж всіх видів, причому – стандартом, що забезпечує передачу мови з дуже гарною якістю. Однак така висока якість передачі мовного сигналу (що є еталоном при оцінці якості інших схем кодування) досягнута в системах ІКМ за рахунок явно надлишкова, при сучасному рівні технології, швидкості передачі інформації [3].

Щоб зменшити властиву ІКМ надмірність і знизити вимоги до смуги пропускання, послідовність чисел, отримана в результаті перетворення мовного аналогового сигналу в цифрову форму, піддається математичним перетворенням, що дозволяють зменшити необхідну швидкість передачі. Ці перетворення «сирого» цифрового потоку в потік меншої швидкості називають «стиском» (а часто – кодуванням, розглядаючи ІКМ як якусь відправну точку для подальшої

обробки інформації). Існує безліч підходів до «стиску» мовної інформації; всі їх можна розділити на три категорії: кодування форми сигналу (waveform coding), кодування вихідної інформації (source coding) і гібридне кодування, що представляє собою сполучення двох підходів.

Кодування форми сигналу. Імпульсно-кодова модуляція, по суті, і являє собою схему кодування форми сигналу. Однак цікавлять більш складні алгоритми, що дозволяють знизити вимоги до смуги пропускання. Розглянуті методи кодування форми сигналу використовують ту обставину, що між випадковими значеннями декількох послідовних обчислень існує деяка залежність. Це дозволяє з досить високою точністю пророчити значення будь-якого відліку на основі значень декількох попередніх йому обчислень.

При побудові алгоритмів кодування названа закономірність використовується двома способами. По-перше, є можливість змінювати параметри квантування залежно від характеру сигналу. У цьому випадку крок квантування може змінюватися, що дозволяє певною мірою згладити протиріччя між зменшенням числа біт, необхідних для кодування величини відліку при збільшенні кроку квантування, і звуженням динамічного діапазону кодера, неминучим без адаптації. Деякі алгоритми передбачають зміну параметрів квантування приблизно в рамках вимовних складів, а деякі змінюють крок квантування на основі аналізу статистичних даних про амплітуду сигналу, отриманих за відносно короткий проміжок часу. По-друге, існує підхід, називаний диференціальним кодуванням або лінійним пророкуванням. Замість того, щоб кодувати вхідний сигнал безпосередньо, кодують різницю між вхідним сигналом і «передвіщеною» величиною, обчисленою на основі декількох попередніх значень сигналу. Описаний метод називається лінійним пророкуванням, тому що він використовує тільки лінійні функції попередніх обчислень. Найпростішою реалізацією останнього підходу є так звана дельта-модуляція (ДМ), алгоритм якої передбачає кодування різниці між сусідніми обчисленнями сигналу тільки одним інформаційним бітом, забезпечуючи

передачу, по суті, тільки знака різниці.

Алгоритмом, побудованим на описані вище принципах, є алгоритм адаптивної диференціальної імпульсно-кодової модуляції (АДІКМ) (G.726). Алгоритм передбачає формування сигналу помилки пророкування і його наступне адаптивне квантування. При досить гарних характеристиках алгоритму, АДІКМ практично не застосовується для передачі мови по мережах з комутацією пакетів, тому що цей алгоритм дуже чутливий до втрат цілих блоків відліку, що відбуваються при втратах пакетів у мережі. У таких випадках порушується синхронізація кодера й декодера, що приводить до катастрофічного погіршення якості відтворення мови навіть при малій імовірності втрат [11-27].

Перелік вихідних даних

У ході розробки корпоративної системи зв'язку з використанням ІР-технологій визначено, що вихідною інформацією є мова або відеозображення на пункті отримання інформації. Як може здатися на перший погляд, вузькополосне кодування мови, що вимагає обчислювальної потужності, є самим складним завданням, виконуваної устаткуванням ІР-телефонії. Однак це не так: алгоритми кодування мови стандартизовані й відмінно документовані, більше того, на ринку доступні досить ефективні їхні реалізації для всіх популярних DSP-платформ. З іншого боку, в устаткуванні ІР -телефонії повинні бути реалізовані багато інших функцій, спосіб реалізації яких не є об'єктом стандартизації.

На передавальній стороні устаткування ІР-телефонії працює за принципом «закодував, передав і забув». На прийомній стороні все набагато складніше. Пакети приходять із мережі із затримкою, що міняється за випадковим законом. Більше того, пакети можуть прийти не в тій послідовності, у якій були передані, а деякі пакети можуть взагалі бути загублені. Приймач повинен справлятися з усіма цими труднощами, забезпечуючи на виході нормальний звуковий потік з тактовою синхронізацією, або генерованим на основі прийнятого потоку даних, або одержуваним із ТфОП по каналах Е1. Прив'язка мовних потоків до місцевого тактового синхросигналу здійснюється шляхом непомітної на слух деформації

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

періодів мовчання у відтвореному сигналі. До цього залишається додати необхідність передачі факсимільної інформації в реальному часі з автоматичним розпізнаванням сигналів факсимільних апаратів і передачу DTMF-сигналів з коректним їхнім відновленням у приймачі.

Сигнали багаточастотного набору номера (DTMF) – просто звукові сигнали, передані по телефонному каналі. При передачі їх по цифровій телефонній мережі не виникає ніяких проблем, тому що кодування за допомогою алгоритму G.711 не накладає ніяких обмежень на вид звукових сигналів – це може бути мова, сигнали модему, або тональні сигнали – всі будуть успішно відтворені на приймачі [17-25].

Вузькополосні кодеки, щоб досягти низьких швидкостей передачі, використовують той факт, що сигнал, який вони кодують, представляє саме мову. Сигнали DTMF при проходженні через такі кодеки спотворюються й не можуть бути успішно розпізнані приймачем на прийомній стороні [27].

Коли користувачеві ТфОП потрібно ввести якусь додаткову інформацію у віддалену систему при вже встановленому з'єднанні, необхідно забезпечити можливість надійної передачі DTMF-Сигналів через мережу IP-телефонії. У випадках, коли система, взаємодіючи з користувачем, просто ставить запитання й чекає введення, тривалість і момент передачі сигналу не важливі. В інших випадках система видає користувачеві список і просить його нажати, наприклад, кнопку «#», як тільки він почує потрібну інформацію; тут ситуація більше складна, і необхідна більше точна прив'язка вчасно.

Існуючі методи передачі сигналів DTMF по мережах IP-телефонії [24,26].

– Обов'язковий метод. Спеціальне повідомлення протоколу H.245 може містити символи цифр і «*», «#». У цьому випадку використовується надійне TCP-з'єднання, так що інформація не може бути загублена. Однак через особливості TCP можуть мати місце значні затримки [24];

– Нестандартний метод. Він може бути застосований у терміналах H.323v2 при використанні процедури fastStart і відсутності каналу H.245. Для передачі

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

сигналів DTMF відкривається спеціальна RTP-сесія, у якій передаються кодовані значення прийнятих цифр, а також дані про амплітуду й тривалість сигналів. Може бути використана та ж сесія, що й для мови, але зі спеціальним типом корисного навантаження. Використання RTP дозволяє прив'язати DTMF- сигнали до реального часу, що є важливою перевагою даного методу [26].

У принципі, перший метод може бути більше кращим, однак у випадку міжнародних викликів і при використанні віддалених систем, що вимагають твердої прив'язки введення користувача до часу, може виявитися необхідним застосувати другий метод. Шлюзи IP-телефонії повинні обов'язково придушувати перекручені сигнали DTMF, що пройшли через основний мовний канал. У протилежному випадку, при відновленні сигналів, про які була прийнята інформація, можуть виникнути неприємні ефекти накладення й розмноження сигналів.

На основі даного огляду функцій устаткування IP-телефонії можна зробити вивід, про тім що, незважаючи на існування стандартних алгоритмів кодування мови, у розроблювачів є величезний простір для діяльності, спрямованої на подальше вдосконалювання технології IP-телефонії.

4.2 Захист розробленого програмного забезпечення

Дані у програмному забезпеченні я захищаю за допомогою NTRU. NTRUEncrypt (аббревіатура Nth-degree TRUncated polynomial ring або Number Theorists aRe Us) – це криптографічна система з відкритим ключем, що раніше називалася NTRU.

Криптосистема NTRUEncrypt, заснована на ґратчастій криптосистемі, створена як альтернатива RSA і криптосистемам на еліптичних кривих (ECC). Стійкість алгоритму забезпечується труднощами пошуку найкоротшого вектора ґрати, що більше стійка до атак, здійснюваним на квантових комп'ютерах. На

відміну від своїх конкурентів RSA, ECC, Elgamal, алгоритм використовує операції над кільцем:

$$\mathbb{Z}[X]/(X^N - 1),$$

усічених багаточленів ступеня, що не перевершує $N - 1$:

$$\mathbf{a}(X) = \mathbf{a} = a_0 + a_1X + a_2X^2 + \dots + a_{N-2}X^{N-2} + a_{N-1}X^{N-1}.$$

Такий багаточлен можна також представити вектором:

$$\vec{\mathbf{a}}(X) = \vec{\mathbf{a}} = \sum_{i=0}^{N-1} a_i X^i = [a_0, a_1, a_2, \dots, a_{N-2}, a_{N-1}].$$

Як і будь-який молодий алгоритм, NTRUEncrypt погано вивчений, хоча й був офіційно затверджений для використання в сфері фінансів комітетом Accredited Standards Committee X9.[1]

Існує реалізація NTRUEncrypt з відкритим вихідним кодом.[2]

NTRUEncrypt, що споконвічно називався NTRU, був винайдений в 1996 році й представлений на конференціях CRYPTO, Конференція RSA, Eurocrypt. Причиною, що послужила початком розробки алгоритму в 1994 році, стала стаття [3], у якій говорилося про легкість злому існуючих алгоритмів на квантових комп'ютерах, які, як показало час, не за горами[4]. У цьому ж році, математики Jeffrey Hoffstein, Jill Pipher і Joseph H. Silverman, що розробили систему разом із засновником компанії NTRU Cryptosystems, Inc. (пізніше перейменованої в SecurityInnovation), Даніелем Лієманом (Daniel Lieman) запатентували свій винахід.[5]

Кільця усічених багаточленів

NTRU оперує над багаточленами ступеня не переважаючої $N - 1$:

$$\mathbf{a} = a_0 + a_1X + a_2X^2 + \dots + a_{N-2}X^{N-2} + a_{N-1}X^{N-1},$$

де коефіцієнти a_0, \dots, a_{N-1} – цілі числа. Щодо операцій додавання й множення за модулем багаточлена $X^N - 1$. Такі багаточлени утворюють кільце R , назване кільцем усічених багаточленів, що ізоморфно кільцю відносин:

$$\mathbb{Z}[X]/(X^N - 1).$$

									Арк.
									61
Вим.	Арк.	№ докум.	Підпис	Дата	ВКРМ-122.21.0094.00.00.ПЗ				

NTRU використовує кільце усічених багаточленів R разом з діленням за модулем на взаємно прості числа p і q для зменшення коефіцієнтів багаточленів.

У роботі алгоритму також використовуються зворотні багаточлени в кільці усічених багаточленів. Слід зазначити, що не всякий багаточлен має зворотний, але якщо зворотний поліном існує, то його легко обчислити.[6][7]

Генерація відкритого ключа

Для передачі повідомлення від Аліси до Боба необхідні відкритий і закритий ключі. Відкритий знають як Боб, так і Аліса, закритий ключ знає тільки Боб, що він використовує для генерації відкритого ключа. Для цього Боб вибирає два «маленьких» поліноми f і g з R . «Малість» поліномів має на увазі в тому розумінні, що він маленький щодо довільного полінома за модулем q : у довільному поліномі коефіцієнти повинні бути приблизно рівномірно розподілені за модулем q , а в малому поліномі вони багато менше q . Малість поліномів визначається за допомогою чисел df і dg :

Поліном f має df коефіцієнтів рівних «1» і $df-1$ коефіцієнтів рівних «-1», а інші – «0». У цьому випадку говорять, що:

$$f \in \mathcal{L}_f$$

Поліном g має dg коефіцієнтів рівних «1» і стільки ж рівних «-1», інші – «0». У цьому випадку говорять, що:

$$g \in \mathcal{L}_g$$

Причина, по якій поліноми вибираються саме таким чином, полягає в тому, що f , можливо, буде мати зворотний, а g – однозначно немає ($g(1) = 0$, а нульовий елемент не має зворотного).

Боб повинен зберігати ці поліноми в секреті. Далі Боб обчислює зворотні поліноми f_p і f_q , тобто такі, що:

$$f \cdot f_p \equiv 1 \pmod{p} \quad \text{і} \quad f \cdot f_q \equiv 1 \pmod{q}$$

Якщо f не має зворотного полінома, то Боб вибирає інший поліном f .

Секретний ключ – це пари (f, f_p) , а відкритий ключ h обчислюється за формулою:

$$h = (pf_q \cdot g) \bmod q.$$

Приклад:

Для приклада візьмемо $df=4$, а $dg=3$. Тоді як поліноми можна вибрати

$$f = -1 + X + X^2 - X^4 + X^6 + X^9 - X^{10}$$

$$g = -1 + X^2 + X^3 + X^5 - X^8 - X^{10}$$

Далі для полінома f шукаються зворотні поліноми за модулем $p=3$ и $q=32$:

$$f_p = 1 + 2X + 2X^3 + 2X^4 + X^5 + 2X^7 + X^8 + 2X^9$$

$$f_q = 5 + 9X + 6X^2 + 16X^3 + 4X^4 + 15X^5 + 16X^6 + 22X^7 + 20X^8 + 18X^9 + 30X^{10}$$

Заключним етапом є обчислення самого відкритого ключа h :

$$h = (pf_q \cdot g) \bmod 32 = 8 + 25X + 22X^2 + 20X^3 + 12X^4 + 24X^5 + 15X^6 + 19X^7 + 12X^8 + 19X^9 + 16X^{10}.$$

Шифрування

Тепер, коли в Алісі є відкритий ключ, вона може відправити зашифроване повідомлення Бобові. Для цього потрібно повідомлення представити у вигляді полінома m з коефіцієнтами за модулем p , обраними з діапазону $(-p/2, p/2]$. Тобто m є «малим» поліномом за модулем q . Далі Алісі необхідно вибрати інший «малий» поліном r , що називається «сліпучої», обумовлений за допомогою числа dr :

Поліном r має dr коефіцієнтів рівних «1» і стільки ж рівних «-1», інші – «0». У цьому випадку говорять, що:

$$r \in \mathcal{L}_r.$$

Використовуючи ці поліноми, зашифроване повідомлення виходить за формулою:

$$e = (r \cdot h + m) \bmod q.$$

При цьому кожній, хто знає (або може обчислити) осліплюючий поліном r , зможе прочитати повідомлення m .

Приклад:

Припустимо, що Аліса хоче послати повідомлення, представлене у вигляді полінома:

$$m = -1 + X^3 - X^4 - X^8 + X^9 + X^{10},$$

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

і вибрала «осліплюючий» поліном, для якого $dr=3$:

$$r = -1 + X^2 + X^3 + X^4 - X^5 - X^7.$$

Тоді шифротекст e , готовий для передачі Бобові буде:

$$e = (r \cdot h + m) \bmod 32 = 14 + 11X + 26X^2 + 24X^3 + 14X^4 + 16X^5 + 30X^6 + 7X^7 + 25X^8 + 6X^9 + 19X^{10}$$

Розшифрування

Тепер, одержавши зашифроване повідомлення e , Боб може його розшифрувати, використовуючи свій секретний ключ. Спочатку він одержує новий проміжний поліном:

$$a = (f \cdot e) \bmod q.$$

Якщо розписати шифротекст, то одержимо ланцюжок:

$$a = (f \cdot e) \bmod q = (f \cdot (r \cdot h + m)) \bmod q = (f \cdot (r \cdot pf_q \cdot g + m)) \bmod q$$

і остаточно:

$$a = (pr \cdot g + f \cdot m) \bmod q.$$

Після того, як Боб обчислив поліном a за модулем q , він повинен вибрати його коефіцієнти з діапазону $(-q/2, q/2]$ і далі обчислити поліном b , одержуваний з полінома a приведенням за модулем p :

$$b = a \bmod p = (f \cdot m) \bmod p,$$

так як:

$$(pr \cdot g) \bmod p = 0.$$

Тепер, використовуючи другу половину секретного ключа й отриманий поліном b , Боб може розшифрувати повідомлення:

$$c = (f_p \cdot b) \bmod p.$$

Неважко бачити, що:

$$c \equiv f_p \cdot f \cdot m \equiv m \pmod{p}.$$

У такий спосіб отриманий поліном c дійсно є вихідним повідомленням m .

Приклад:

Боб одержав від Аліси шифроване повідомлення e :

$$e = 14 + 11X + 26X^2 + 24X^3 + 14X^4 + 16X^5 + 30X^6 + 7X^7 + 25X^8 + 6X^9 + 19X^{10}$$

Використовуючи секретний ключ f Боб одержує поліном a :

$a = f \cdot e \pmod{32} = 3 - 7X - 10X^2 - 11X^3 + 10X^4 + 7X^5 + 6X^6 + 7X^7 + 5X^8 - 3X^9 - 7X^{10} \pmod{32}$,
з коефіцієнтами, що належать проміжку $(-q/2, q/2]$. Далі перетворить поліном a у
поліном b , зменшуючи коефіцієнти за модулем p .

$$b = a \pmod{3} = -X - X^2 + X^3 + X^4 + X^5 + X^7 - X^8 - X^{10} \pmod{3}$$

Заключний крок – перемноження полінома b із другою половиною
закритого ключа f_p :

$$c = f_p \cdot b = f_p \cdot f \cdot m \pmod{3} = m \pmod{3}$$

$$c = -1 + X^3 - X^4 - X^8 + X^9 + X^{10}$$

Який є вихідним повідомленням, що передавала Аліса.

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Встановлення програмного додатку на ПК здійснюється простим копіюванням програмної папки IP_TEL_NET на жорсткий диск ПК. Нижче наведено вміст вказаної папки:

- *IpTel.exe* – виконавчий файл програми;
- *abc.ttf* – файл зі шрифтом;

Налаштування зв'язку між кінцевими абонентами

1. Після запуску програми на екран виводиться вікно завантаження програмного забезпечення та інформації про автора, яке спливає.

2. Після цього необхідно пересвідчитися у наявності доступу до мережі. Для цього необхідно перевірити наявність мережної карти у персональному комп'ютері та підключеного шнура до мережі. Перевірити системні параметри мережної карти. Зайти в меню Пуск→Налаштування→Мережні підключення→Налаштування мережних підключень: вибрати: IP-адрес, шлюз, маску під мережі, використовуваний протокол (TCP/IP).

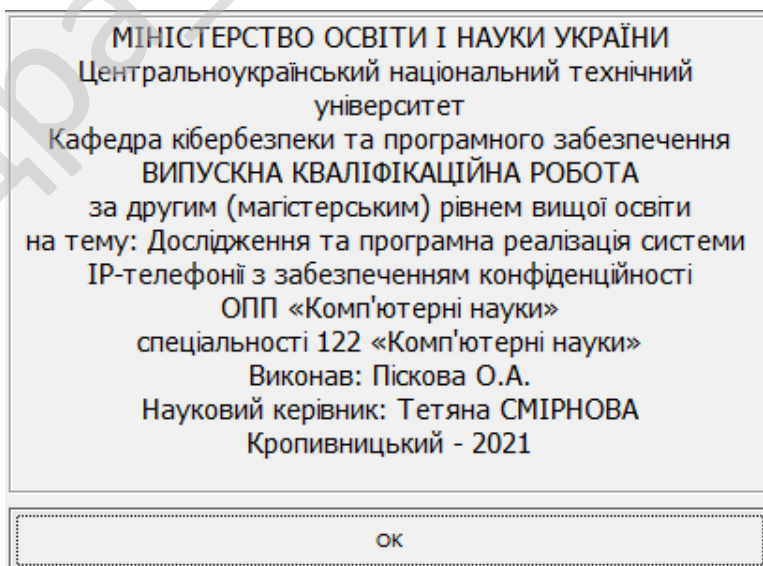


Рисунок 5.1 – Вікно авторського права

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

На мережній карті повинна горіти зелена лампочка. Це каже про те, що фізично мережна карта працездатна та готова до роботи з мережею.

3. Потім необхідно перевірити параметри установки файрволу:

– системи дозволу IP-портів;

– привілеї користувача;

– поточний статус файрволу та перевірити наявність на одночасну роботу декількох файрволів на персональному комп'ютері.

4. Перевірити на наявність вірусів антивірусом та файрволом, або включити максимальний рівень захисту від вірусів. Антивіруси забезпечують найвищий рівень захисту. Виявляються й видаляються всі типи вірусів і троянських програм, ворожі об'єкти Java/Active, блокується доступ до небезпечних WEB-ресурсів. Захист від вірусів охоплює всі операційні системи й групові додатки, які використовуються в сучасних корпоративних мережах: клієнтські ОС Windows 10, Macintosh; серверні ОС Windows Server, FreeBSD, Linux, HP-UX, AIX, SCO, Solaris; групові додатки MS Exchange 5.5/2000 і Lotus Notes/Domino; інтернет-шлюзи Windows NT/2000 і Sun Solaris; мережні пристрої зберігання даних NetApp; ОС мікрокомп'ютерів (PDA) Palm OS, Windows CE / Pocket PC, EPOC (Psion).

Антивірус призначений для захисту від вірусів файл-серверів. Файлові сервери – одне із самих уразливих місць корпоративної мережі: якщо поразка вірусом робочого місця приведе до відмови одного комп'ютера, то зараження файл-сервера може порушити роботу всієї мережі. Крім того, заражений сервер може зберігати працездатність, будучи джерелом вірусів для користувачів мережі. Антивірус встановлюється на сервер і у фоновому режимі перевіряє всі файли, до яких відбувається звертання від клієнтів. Він забезпечується останніми версіями антивірусних баз, автоматично роблячи відновлення через Інтернет або по локальній мережі. Крім того, антивірус здатний самостійно здійснювати регулярні перевірки дисків за розкладом, не вимагаючи участі адміністратора

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

Файрвол контролює весь трафік на робочій станції, дозволяючи тільки відомим додаткам підключатися до комп'ютера, блокуючи небезпечний трафік, що генерований хакерами, шкідливим кодом, агентами розподілених атак типу "відмова в обслуговуванні", уразливими або неавторизованими додатками й іншими типами атак, що піддають погрозі збереження даних і працездатність мережі. Адміністратор мережі може настроїти роботу файрвола прозоро для користувачів, так що користувач навіть не буде знати про присутність на його машині брандмауера.

Антивірусний монітор, повинен працювати в режимі реального часу. Всі операції читання або запису файлів відбуваються після перевірки їх на предмет вірусів. Така організація дозволяє автоматично захищати не тільки сервер від клієнтів, але й клієнтів від сервера, у випадку якщо він заражений вірусами. Заражені файли автоматично виліковують, видаляються або переміщуються в карантинну зону, після чого адміністраторові відсилається повідомлення.

Сканер вірусів може бути використаний для ручної примусової перевірки дисків на предмет вірусів. За допомогою автоматичного розкладу можна робити регламентні перевірки всіх дисків сервера. Планувати такі операції можна на неробочий час із метою більше ефективного використання обчислювальних ресурсів процесора. Виявлені заражені файли автоматично виліковують, віддаляються або переміщуються в карантинну область, з наступним повідомленням адміністратора.

Антивірусний захист у фоновому режимі захищає робоче місце від можливого зараження вірусами в реальному часі. Він сканує файли, електронну пошту й WEB-ресурси безпосередньо при звертанні до них користувачів. У будь-який момент часу користувач або адміністратор мережі може перевірити локальні або мережні диски на предмет вірусів. Перевірка може бути запущена автоматично за розкладом.

Централізоване керування політикою захисту дозволяє адміністраторові мережі задавати стандартний набір правил (політику) роботи програм і

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

впроваджувати їх по всій мережі підприємства або окремих його частин. Це значно заощаджує час адміністратора, рятуючи його від необхідності відвідувати кожне робоче місце, а також повністю виключає ймовірність зараження вірусами внаслідок некоректного налаштування антивірусних програм користувачами.

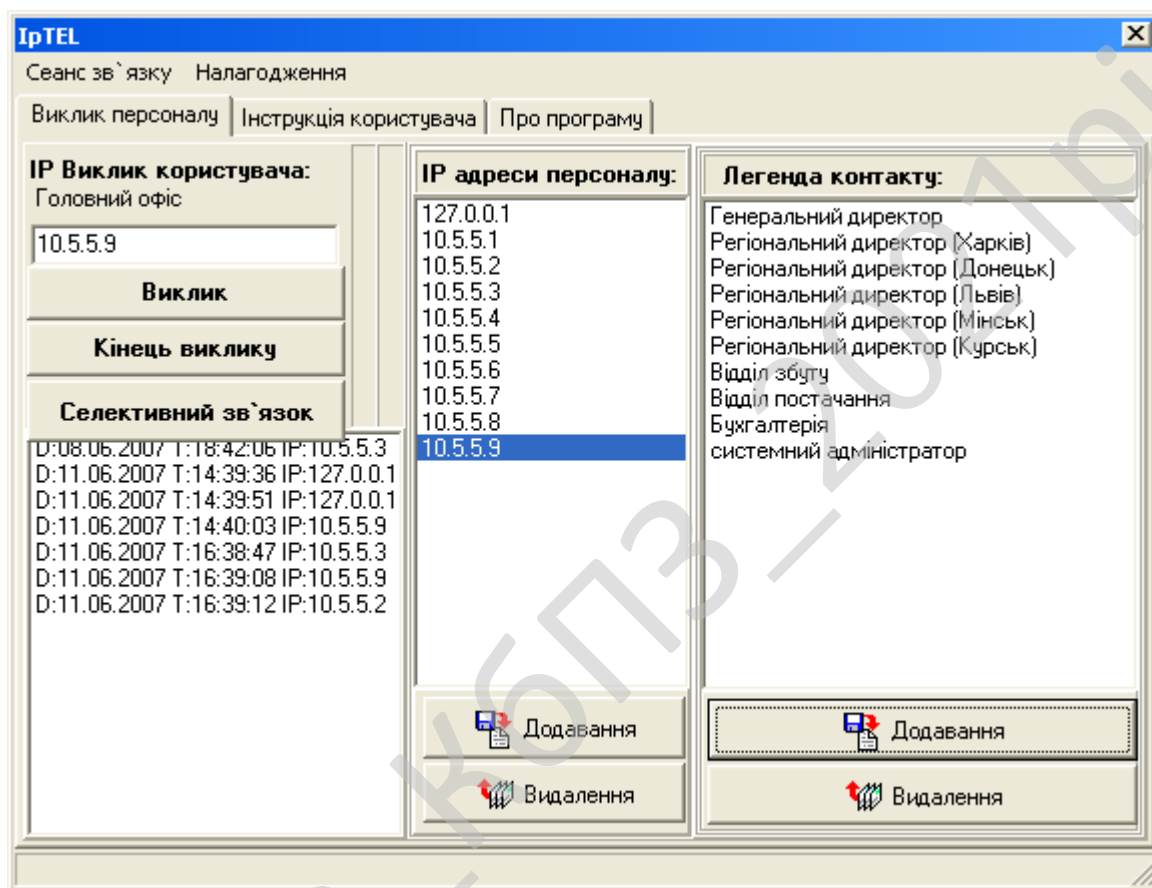


Рисунок 5.2 – Головна форма програми

5. Якщо є IP -телефон, то підключити його згідно інструкції користувача, яка до нього додається.

6. Запустити розроблене у ході виконання магістерської роботи програмне забезпечення на кінцевих персональних комп'ютерах.

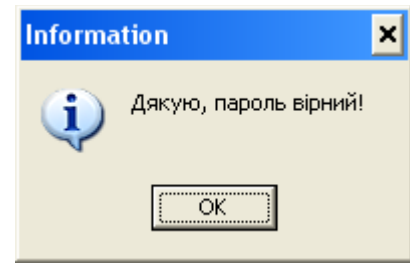
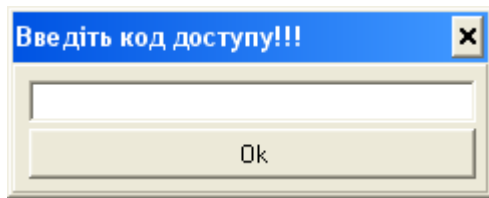


Рисунок 5.3 – Вікно правильного введення паролю

7. Визначити IP-адресу кінцевих машин. Для цього необхідно виконати наступні дії:

- Отримати інформаційні дані від адміністратора локальної мережі.

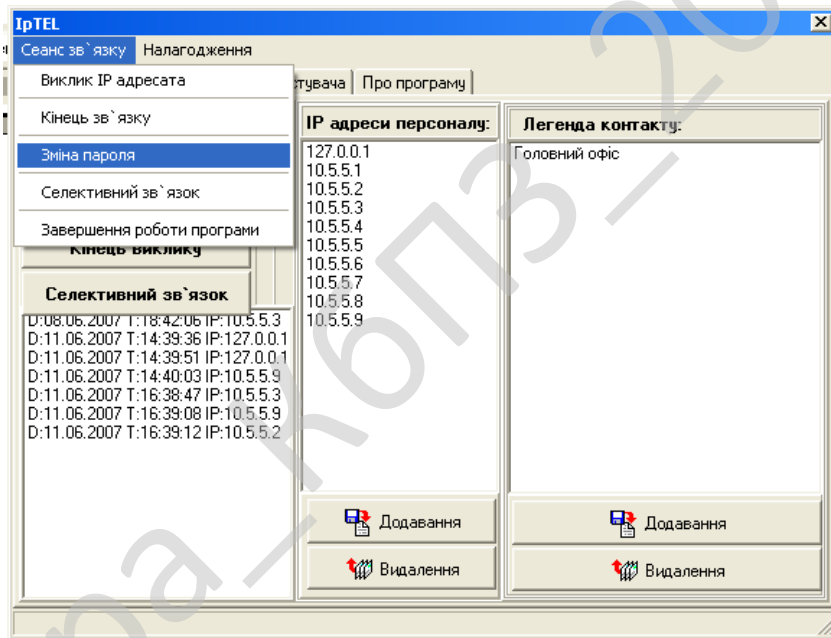


Рисунок 5.4 – Вхідження у режим зміни пароля

– У разі неможливості виконання попереднього пункту необхідно виконати наступні дії: зайти на закладку: Пуск→Налаштування→Мережні підключення вибрати ярлик мережного підключення й натиснути на ньому правою кнопкою миші. У контекстному меню, яке з'явиться вибрати: Властивість→Вибір пункту TCP/IP→в закладці IP-адрес. Там буде відображено IP-адресу персонального комп'ютера у локальній мережі.

8. Заповнити базу даних IP-адрес та легенду до кожної з неї.

9. Після виконання усіх операцій, перерахованих вище, працюємо з розробленим програмним забезпеченням, згідно інструкції користувача.

Вхід в систему

Система захищена програмним чином і її активізація здійснюється через введення пароля у відповідне поле та натисненням кнопки. По замовчуванню встановлено пароль «123». За умови коректного введення пароля (рисунок 5.4) система надає користувачеві відповідний доступ для входу в систему, в іншому випадку програма закінчує свою роботу. Існує можливість зміни паролю. Для цього у вікні зміни паролю необхідно ввести старий пароль, ввести новий пароль та продублювати новий пароль. Фіксація паролю здійснюється кнопкою “Ок”.

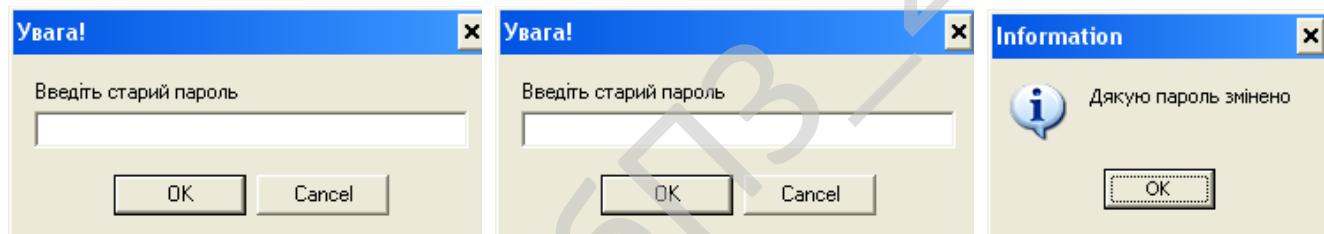


Рисунок 5.5 – Вікно зміни паролю

Організація бази клієнтів

В додатку є можливість редагування IP-адрес та легенди до них.

Зміни вносяться безпосередньо в сітку відповідних таблиць, їх фіксація здійснюється за допомогою кнопки ОК.

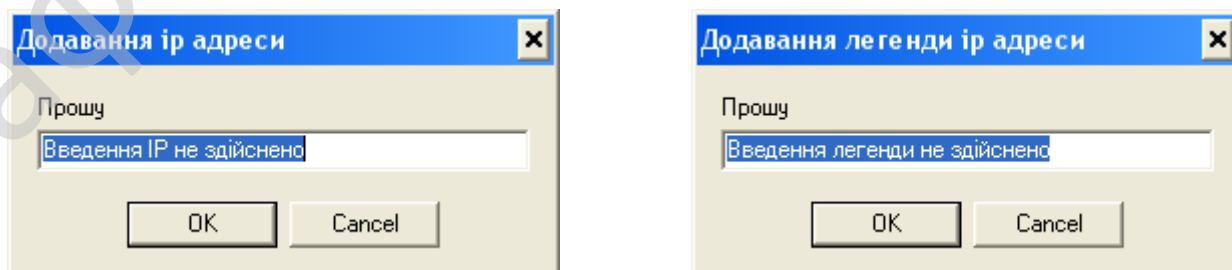


Рисунок 5.6 – Форма редагування адрес та легенд до них

Організація селективного зв'язку

Для організації селективного зв'язку необхідно за натяти клавішу Ctrl та за допомогою лівої кнопки миші виділити ті номери з якими потрібно організувати зв'язок (рисунок 5.7).

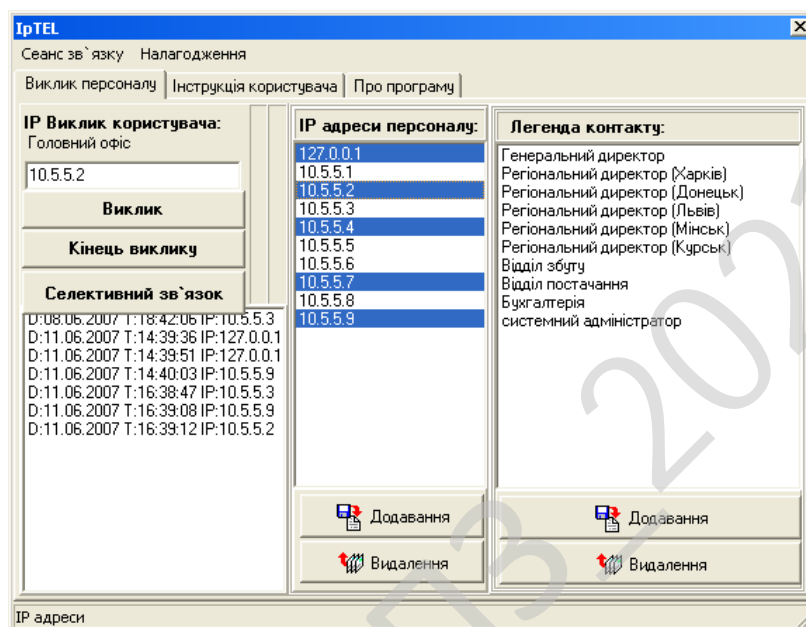


Рисунок 5.7 – Форма вибору номерів для селективного зв'язку

Можливі проблеми та їх усунення

Причинами неможливості доступу мережі можуть бути:

- проблеми з мережею;
- проблеми з мережною картою;
- неактивність сервера БД або некоректність його встановлення.

Перевірку працездатності мережної карти приведена у пункті 5.3.

Перевірка запуску сервера БД здійснюється через запуск менеджера сервера БД із панелі управління Windows

При цьому на ПК не повинно бути файрволів або проксі, які не дають йому приймати з'єднання на порту 3050 по протоколу tcp/ip (або файрвол/проксі потрібно налаштувати, щоб не заважали СУБД Interbase).

Після цього потрібно переконатись, що якщо у вас в мережі немає серверів Netware, то не один клієнт або сервер не має встановленої підтримки протоколів IPX/SPX (NWLink і т.п). Цей протокол не потрібний для роботи.

Взагалі, якщо ОС встановлена коректно, і мережа правильно налаштована, то InterBase або Firebird повинні також функціонувати нормально. З приводу налаштування мережі слід звертатись до її адміністратора або програмної документації ОС.

Причинами помилки можуть бути наступні:

- клієнтська частина (gds32.dll, fbclient.dll) не відповідає версії сервера.
- клієнтська частина не підтримує локальний протокол взагалі (наприклад, в Firebird 1.5.1 for Windows, Classic). Також локальний протокол не працює, якщо на Win2003 або WinXP активізовані служби Terminal Services.
- особливості конкретного логіна або версії операційної системи.

У всіх випадках проблем з локальним протоколом рекомендується перевірити всі вище перераховані варіанти, і якщо їх не вдалось усунути – використовувати локальний мережевий протокол для з'єднання з БД.

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи IP-телефонії з забезпеченням конфіденційності.

Метою розробки є дослідження та програмна реалізація системи IP-телефонії з забезпеченням конфіденційності.

Об'єктом дослідження є процес IP-телефонії з забезпеченням конфіденційності.

Предметом дослідження є методи IP-телефонії з забезпеченням конфіденційності.

Методи дослідження базуються на методах теорії передачі мультимедійних даних, методах математичної статистики, методах розробки програмного забезпечення.

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

–Удосконалено метод IP-телефонії з забезпеченням конфіденційності.

–Розроблено вітчизняний продукт IP-телефонії з забезпеченням конфіденційності, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

7 ДАНІ ПРО ЕКОНОМІЧНУ ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ

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

Після ознайомлення з підприємством та засобами розробки програмної продукції був розроблений план розробки програми. Був підрахований необхідний час для розробки та впровадження програми. Цей час склав 60 днів (три місяці).

В магістерській роботі було проведено дослідження та виконана програмна реалізація системи IP-телефонії з забезпеченням конфіденційності.

Розроблене програмне забезпечення має достатню надійність і задовольняє усім поставленим умовам, а саме:

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

Таблиця 7.1 – Початкові дані

Показники	Позначення	Характеристика або величина
1	2	3
1. Кількість розроблених програм період, шт.	N	1
2. Кількість екземплярів програм, шт.	Ne	24 (варіант 94)
3. Запланований термін розробки, днів	Fpq	60 (3 місяці)
4. Група задачі підсистеми управління (1-6)	–	1
5. Ступінь новизни задачі (А, Б, В, Г)	–	Б
6. Складність алгоритму (1, 2, 3)	–	2

Продовження таблиці 7.1

1	2	3
7. Кількість макетів вхідної інформації	–	3
8. Кількість форм вихідної інформації	–	4
9. Мова програмування (1-6)	–	2
10. Попередній досвід (1-6)	–	3
11. Гнучкість проекту ПП (1-6)	–	3
12. Детальність проекту ПП (1-6)	–	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

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-122.21.0094.00.00.ПЗ

Арк.

76

Продовження таблиці 7.1

1	2	3
28. Досвід роботи з мовою і інструментами середовища розробки (1-6)	–	2
29. Досвід роботи з програмними інструментами розробки (1-6)	–	3
30. Розробка ПЗ для декількох серверів одночасно (1-6)	–	2
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн.	–	24000
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: A – коефіцієнт Боема, $A = 2,45$;

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

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.

$$B = 1,01 + 0,001(2,43 + 3,64 + 3,38 + 3,95 + 2,73) = 1,027.$$

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

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

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

де: $\prod V_j$ – 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:

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

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

$$T_{РП} = 0,3 \cdot 2,66 \cdot 9,37^{0,33+0,2(1,026-1,01)} \cdot 100 = 168 \text{ люд/день.}$$

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

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

Таблиця 7.2 – Визначення трудомісткості розробки програмного забезпечення

Стадії розробки	Трудомісткість за типовими нормами та розрахунками	
	Величина, люд/дні	Підстава
Технічне завдання	9	Д5
Ескізний проект	10	Д6
Технічний проект	9	Д7
Робочий проект	168	Ф 7.1-7.4
Впровадження	13	Д13
Всього	209	–

7.3 Визначення чисельності виконавців і планового фонду зарплати

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

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

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

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

$$Ч = \frac{209 \cdot 1}{60 - 5} = 3,8 \text{ ставки.}$$

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

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

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

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

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

Виходячи з цього фонд робочого часу інженерів-електронщиків складає:

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

$$\Phi_{op}^c = \frac{227 \cdot 3}{1,2} = 567,5 \text{ год.}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		80

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

$$Ч_{ел} = 567,5 / (60 \cdot 8) = 1,2 \text{ ставки.}$$

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, серверу доступу ADSL (OC Linux), налаштування ADSL, VPN PPPoE, Frame Relay, Wi-Fi	2	0,5
	Налаштування і конфігурування базової станції безпроводного зв'язку (CMTS)	0,5	
	Розробка та впровадження проектів з організації зв'язку між віддаленими об'єктами, ЛОМ	0,5	
	Забезпечення цілодобової роботи зв'язку клієнтів до мережі Інтернет	1	
Всього		4	

Продовження таблиці 7.4

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

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-122.21.0094.00.00.ПЗ

Арк.

82

Складемо штатний розклад виконавців.

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньомісячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	1	13000	39000
Продакт-менеджер	0,25	12000	9000
Інженер-програміст	3,8	12000	136800
Інженер-електронщик	1,2	11500	41400
Інженер-системотехнік	0,25	11500	8625
Адміністратор мережі	0,5	11500	17250
Системний програміст	0,25	11500	8625
Дизайнер WEB	0,25	12000	9000
Інженер-верстальник	0,25	11700	8775
Бухгалтер-економіст	0,5	12500	18750
Всього за період розробки	$R_{cn} = 8,25$	-	$\Phi_{роб} = 297225$

Розрахуємо середньоденну зарплату одного виконавця:

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

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

$$z_{cd} = \frac{297225}{8,25 \cdot 60} = 600 \text{ грн.}$$

7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника

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

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83

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

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

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

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

Згідно даних ТОВ науково-дослідницького консалтингового підприємства «Пектораль» (м. Кіровоград, вул. Глинки 16) ціна одного квадратного метра площі новобудови, вік якої не перевищує 25 років, по місту складає 800...1600 у.о./ m^2 . Враховуючи, що курс складає 1 у.о. = 25 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:

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

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

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

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 за прайсом фірми Brain за 24.10.21 – джерело <http://brain.com.ua>.

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Персональний комп'ютер		11771
Системний блок		7771
Процесор	AMD A8-3870K 3.0 GHz Socket FM1 Quad Core	2500
Системна плата	ASUS F1A55-M , AMD A55 , Socket FM1	1100
Жорсткий диск	HDD 500 Gb SAMSUNG Barracuda HD502HJ (3.5", 500ГБ, 16МБ, SATA II-300)	1290
Оперативна пам'ять	DIMM 4096Mb DDR3 PC3-12800 Kingston 1600MHz, 512M x 64, CL9-9-9-27, 1.65V w/heatsink, HyperX	834
DVD-привод	DVD±RW ASUS DRW-24B5ST Black Bulk	416
Корпус	Logicpower 8702 - 550w 12cm	1411
Кардрідер внутрішній	Transcend TS-RDF8K USB 3.0	220
інше	Клавіатура, мишка	Подаруно
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 прийняті в межах до 10% від оптової ціни.

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

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	20	-
5. Транспортні засоби	143000	25	
6. Господарський інвентар	28000	20	-
Всього по групах 5, 6	176190	-	35238
7. Нематеріальні активи	24000	10	2400
Разом	$K_p = 1733406$		$A_p = 170646$

Примітка: вартість автомобіля Sens (Standard+) взята по даним з автосалону «Кіровоград-Авто», джерело <http://kirovograd-avto.ukravto.ua/catalog/tm-9/model-80/description>, 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.

$$Z_o = 600 \cdot 209 / 24 = 5225 \text{ грн.}$$

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

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		87

$$Z_{\partial} = Z_o \cdot H_q \cdot 0,01, \quad (7.12)$$

де: H_q – норматив додаткової зарплати, %.

$$Z_{\partial} = 5225 \cdot 10 \cdot 0,01 = 523 \text{ грн.}$$

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

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

де: H_c – відрахування на соціальні потреби, %.

$$C_{oc} = 0,01 \cdot 22(5225 + 523) = 1265 \text{ грн.}$$

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, %.

$$G_{ocn} = 5225 \cdot 15 \cdot 0,01 = 784 \text{ грн.}$$

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 виданих викладачем норм приймаємо одну пачку паперу на місяць розробки. Тоді, враховуючи, що вартість пачки паперу складає $C_n = 105$ грн., визначаємо вартість паперу за період розробки $N_M = 3$ міс:

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

$$Z_{M1} = 105 \cdot 3 = 315 \text{ грн.}$$

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

						БКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			88

$$З_{M2} = \sum C_{\partial}, \quad (7.17)$$

де: C_{∂} – вартість дисків CD/DVD: CDR TDK 700Mb, 80Min, 52x Cake box – 12 грн./шт., DVD-R LG 4,7Gb, 16x speed Cake box – 12 грн./шт.

$$З_{M2} = 12 \cdot 24 + 12 = 300 \text{ грн.}$$

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

$$З_{M3} = \sum C_{3.}, \quad (7.18)$$

де: $C_{3.}$ – вартість розхідних матеріалів друкуючих пристроїв: відновлення та заправка картриджу для Canon i-SENSYS LBP6030W – 574 грн.; картридж для Epson Stylus Photo P50 – 558 грн.; відновлення картриджу для MF217W – 570 грн.

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

$$З_M = (315 + 300 + 1702) / 24 = 97 \text{ грн.}$$

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

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

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

$$O_n = 5225 \cdot 15 \cdot 0,01 = 784 \text{ грн.}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_e = 24$ 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.

$$A_m = 170646 \cdot 3 / (24 \cdot 12) = 1777 \text{ грн.}$$

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

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

$$C_n = 5225 + 523 + 1265 + 784 + 97 + 784 + 1777 = 10455 \text{ грн.}$$

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		89

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

Для даного програмного забезпечення рівень рентабельності складає 50%.

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

де: P_n – рівень рентабельності, %.

$$P_p = 0,01 \cdot 50 \cdot 10455 = 5228 \text{ грн.}$$

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

Таблиця 7.9 – Нормативна калькуляція собівартості розробки програмного забезпечення задачі

Найменування статей витрат	Позначення	Величина, грн
1. Основна зарплата виконавців	Z_o	5225
2. Додаткова зарплата виконавців	Z_d	523
3. Відрахування на соціальні потреби	C_{oc}	1265
4. Загальногосподарські витрати	G_{ocn}	784
5. Витрати на матеріали	Z_M	97
6. Освоєння нових операційних систем, мов програмування	O_n	784
7. Амортизація основних фондів	A_m	1777
8. Повна собівартість програмного забезпечення	C_n	10455
9. Плановий прибуток	P_p	5228
10. Ціна підприємства $C_n = C_n + P_p$	C_n	15683
11. Податок на додану вартість $ПДВ = 0.01 \cdot H_{об} \cdot C_n$	$ПДВ$	3136,6
12. Відпускна ціна програмної продукції $C = C_n + ПДВ$	C	18819,6

Вим.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

ВКРМ-122.21.0094.00.00.ПЗ

Арк.

90

7.6 Визначення об'єму капітальних вкладень у споживача програмної продукції

Об'єм капітальних вкладень у споживача програмної продукції визначаємо на основі балансової вартості основних фондів, яка враховує ціну, транспортно-заготівельні витрати, вартість будівель, монтажних та пусконаладжувальних робіт, а також витрати на випробування у виробничих умовах. Результати розрахунків зводимо у таблицю 7.10.

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

Найменування капітальних вкладень	Сума за варіантами, грн.	
	Базовий	Новий
Вартість програмної продукції	–	18820
Всього капітальних витрат	–	18820

7.7 Визначення експлуатаційних витрат

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

$$Z_o = C_e O_e M_e (1 + 0,01H_q)(1 + 0,01H_c), \quad (7.23)$$

де C_e – чисельність працівників, чол.;

O_e – заробітна плата, грн./год;

M_e – час, що витрачається на нарахування ЗП.

Після купівлі нового програмного забезпечення час на нарахування зменшився з 630 годин на рік до 393 годин на рік, тому витрати на технічне обслуговування зменшилися з:

$$Z_{o_{баз}} = 1 \cdot 36 \cdot 630 \cdot 1,1 \cdot 1,22 = 30437 \text{ грн.},$$

до:

$$Z_{o_{нов}} = 1 \cdot 36 \cdot 393 \cdot 1,1 \cdot 1,22 = 18987 \text{ грн.}$$

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

Таблиця 7.11 – Розрахунок експлуатаційних витрат у споживача програмної продукції

Найменування статей витрат	Позначення	Сума витрат за варіантами, грн.	
		Базовий	Новий
1. Заробітна плата(основна, додаткова, відрахування в соціальні фонди)	Z_p	30437	18987
2. Витрати на електроенергію	$Z_{ел}$	9639	8925
3. Витрати на амортизацію	$Z_{ам}$	0	4705
Всього витрат за рік	I	40076	32617

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

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

$$Z_{ел\ баз} = 0,54 \cdot 8500 \cdot 2,1 = 9639 \text{ грн.}$$

$$Z_{ел\ нов} = 0,50 \cdot 8500 \cdot 2,1 = 8925 \text{ грн.}$$

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

Таблиця 7.12 – Розрахунок амортизаційних відрахувань

Групи основних фондів	Норма амортизації %	Балансова вартість, грн., за варіантами		Сума відрахувань, грн за варіантами	
		Базовий	Новий	Базовий	Новий
Програмна продукція	25	–	18820	–	4705
Всього відрахувань	-	–	18820	–	4705

7.8 Визначення економічної ефективності програмної продукції

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

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

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

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

$$E_e = (15683 - 10455) \cdot 24 - (0,05 \cdot 1408000 + 0,5 \cdot 125216 + 0,25 \cdot 33190 + 0,2 \cdot 143000 + 0,1 \cdot 24000) \cdot 3/12 = 82396 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у виробника програмної продукції:

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

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

$$T_e = \frac{325406}{(15683 - 10455) \cdot 24 \cdot 12 / 3} = 0,7 \text{ років.}$$

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

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

де: $I_{\bar{o}}$, I_n – величина експлуатаційних витрат за базовим и новим варіантом відповідно; $K_{\bar{o}}$, K_n – об'єм капітальних вкладень за варіантами, що порівнюються.

$$E_{cn} = (40076 - 32617) - 0,25 \cdot 18820 = 2754 \text{ грн.}$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

									Арк.
									93
Вим.	Арк.	№ докум.	Підпис	Дата	ВКРМ-122.21.0094.00.00.ПЗ				

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1. Кількість екземплярів програми	Прим.	24
2. Повна собівартість розробленої програми	Грн.	10455
3. Ціна розробленої програми	Грн.	15683
4. Плановий прибуток від реалізації розробленої програми	Грн.	5228
5. Рентабельність програмної продукції	%	50
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	1733406
7. Загальний прибуток від реалізації програмної продукції	Грн.	125472
8. Величина економічного ефекту при виготовлені програмної продукції	Грн.	82396
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років	0,7
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	18820
11. Величина економічного ефекту у користувача програмної продукції	Грн.	2754
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Рік	2,6

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

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

$$T_{cn} = \frac{18820}{40076 - 32617} = 2,6 \text{ років.}$$

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.

Кафедра _ КБПЗ _ 2021 рік

					VKPM-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		95

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Протягом усієї історії людство приділяє прискіпливу увагу безпеці життя [1]. Охорона праці є складовою частиною безпеки життя.

Законом України “Про охорону праці” [2] 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», НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин», та ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин».

Програмісти у процесі роботи мають негативний вплив на органи зору, а також мають значну розумову напругою 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.

При розгляді шкідливих чинників роботи програмістів та інших спеціалістів ІТ будемо керуватись наступними нормативно-правовими актами: «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98, та

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		96

«Правила охорони праці під час експлуатації електронно-обчислювальних машин» НПАОП 0.00-1.28-10,

Умови праці програміста включають наступні фактори:

- параметри повітряного середовища в приміщенні;
- вентиляція приміщення;
- освітлення приміщення;
- параметри повітряного середовища в приміщенні, тощо.

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

8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

Програміст працює з електронно-обчислювальною машиною (ЕОМ) та іншим обладнанням, яке є джерелом небезпеки ураження електричним струмом. Так як робота програміста характеризується істотним зоровим навантаженням, то вимагає належного освітлення. Так як програміст постійно перебуває в приміщенні, тому для комфортних умов праці в цьому приміщенні необхідно створити належний мікроклімат.

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;
- монотонність праці;

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

- електромагнітні (у т.ч. високочастотні) електромагнітні випромінювання (коливання);
- несприятливі мікрокліматичні умови;
- нервово-емоційна напруженість праці;
- інтелектуальні навантаження;
- невідповідність ергономічних показників робочого місця діючим вимогам;
- шуми;
- статичні навантаження на кістково-м'язовий апарат;

8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста

Розглянемо умови праці у приміщенні, в якому працюють програмісти. Геометричні розміри приміщення наведено у таблиці 8.1.

Таблиця 8.1 – Розміри приміщення

Найменування	Значення, м
Ширина	4
Довжина	5,5
Висота	2,8

Таблиця 8.2 – Площа та обсяг приміщення, на одного працюючого

Геометрична характеристика	Одиниця виміру	Нормативне значення*	Фактичне значення
Площа, S	м ²	не менше 6.0	7,3
Обсяг, V	м ³	не менше 20.0	20,5

* Згідно ДСанПіН 3.3.2.007-98 (Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин).

У зазначеному приміщенні працює 3 людей. За даними, які наведено у табл. 8.1 та табл. 8.2, можна зробити висновок, що площа та об'єм приміщення у розрахунку на одно робоче місце програміста відповідають нормативним вимогам (Наказу Міністерства соціальної політики України № 207, від 14.02.2018 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями», ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» та НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин»).

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, яка виконується в даному приміщенні, відноситься до категорії Іа. В цьому випадку людина витрачає енергії до 120 ккал у годину. Вологість повітря у 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 Іа, так і розглянутого приміщення. У

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		99

приміщеннях, де встановлено ЕОМ, рекомендується застосування тільки оптимальних значень показників мікроклімату.

Таблиця 8.3 – Оптимальні і фактичні значення параметрів мікроклімату

Пора року	Оптимальні для Ia			Фактичні		
	Температура, °C	Воло- гість,%	Швидкість повітря, м/с	Температура, °C	Воло- гість%	Швидкість повітря, м/с
Холодна	22-24	40-60	0,1	22-23	42-57	0,11
Тепла	23-25	50-70	0,1	23-24	55-70	0,1

Проведений аналіз показує, що показники мікроклімату в приміщенні відповідають установленим нормам. Штучне опалення застосовується у холодний період року.

В літню пору застосовується кондиціонер.

Для боротьби з пилом робляться регулярні провітрювання та вологі прибирання приміщенні.

У приміщенні знаходяться наступні джерела шуму: принтер *HP Laser 135a*, електродвигуни вентиляторів ЕОМ.

Одним з найважливіших факторів, які впливають на ефективність трудової діяльності людини, та попереджають травматизм і професійні захворювання програмістів є освітлення на робочому місці.

Працю працівника, який постійно працює за комп'ютером, згідно ДБН В.2.5 – 28 – 2006 р. можна віднести до роботи з малою точністю (найменший розмір об'єкта розрізнення від 1 до 5 мм) V-го розряду зорової роботи, з великою контрастністю об'єкта розрізнення (символів на екрані дисплея), з темним тлом (під розряд зорової роботи B). Приміщення можна віднести до 1-ої групи приміщень, у яких проводиться розрізнення об'єктів зорової роботи при фіксованому напрямку лінії зору того, що працює на робочу поверхню. Для такого типу приміщень і розряду зорової роботи 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. 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'язковою наявністю масок-клапанів, або іншого спорядження для штучного дихання. Регулярна періодична перевірка параметрів заземлення та занулення (вимірювання опору ланцюга).

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		101

8.5 Розрахункова частина

Початкові данні для розрахунку:

Тип заземлення: робоче заземлення нульової точки трансформатора. Заземленню підлягає виробниче обладнання організації. Напруга – 220/380 В. Розташування заземлюючих електродів – по контуру.

Розрахунок проводиться за допустимим опором розтіканню струму заземлювача методом коефіцієнта використання заземлювачів.

Для заземлення застосовуються: вертикальні електроди – прутки довжиною $L = 2$ м. Відстань між вертикальними заземлювачами (електродами) $A=4$ м. Діаметр вертикального електрода (прутка) $D=18$ мм, Тип горизонтального заземлювача: металева полоса. Розміри перетину з'єднуючої полоси: 60 x 5 мм. ($b=60$ мм.). Глибина закладення контура заземлення $t= 0,7$ м. Товщина верхнього шару ґрунта $H=1,8$ м. Відстань від центра вертикального заземлювача до поверхні замі: $T=1,7$ м. Сезонний кліматичний коефіцієнт вертикального заземлювача: $C_v=1,2$. Сезонний кліматичний коефіцієнт горизонтального заземлювача: $C_v=1,5$. Питомий опір верхнього шару ґрунта (чернозем): $\rho_1= 200$ Ом*м. Питомий опір нижнього шару ґрунта (пластичний суглинок): $\rho_2= 30$ Ом*м. Опір заземлювача, який нормується: $R_{ЗН} = 4$ Ом. Коефіцієнт використання вертикального заземлювача $\eta_C = 0,69$ (табличне значення, залежить від кількості вертикальних стержнів та співвідношення A/L). Коефіцієнт використання горизонтального заземлювача $\eta_H = 0,43$ (табличне значення, залежить від співвідношення A/L).

Розрахунок захисного заземлення можна автоматизувати за допомогою програми, сирцевий код якої опублікован на стр.13-16 [3], або будь якої іншої, наприклад, програми «Електрик» (автор Сумеркін Є.О.), URL: <http://rzd2001.narod.ru>. Остання була використана для автоматизації процесу перевірки розрахунку.

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		102

Розрахунок.

Еквівалентний питомий опір:

$$\rho_{\text{екв}} = \psi \rho_1 \rho_2 L / [\rho_1 \psi(L-H+t) + \rho_2 \psi(H-t)] = 57,83 \text{ Ом.}$$

Опір одного вертикального заземлювача:

$$R_0 = [\rho_1 / (2\pi * L)] * [(\ln) 2L/D + 0,5 \ln[(4T+L)/(4T-L)]] = 26,28 \text{ Ом.}$$

Коефіцієнт заземлення при $\rho_{\text{екв}}$ менше, ніж $100 \text{ Ом} * \text{м}$ (у нас $\rho_{\text{екв}} = 57,83 < 100$) = 1, при цьому опір, який нормується $R_{\text{норм}} = 4 \text{ Ом}$. [3].

Опір розтіканню горизонтального заземлювача (полоси):

$$R_{\Pi} = 0,366(\rho_{\text{екв}} \psi / L_{\Pi} \cdot \eta_{\Pi}) \lg(2 / L_{\Pi} * L_{\Pi} / bt) = 6,25 \text{ Ом,}$$

де мінімальна довжина горизонтального заземлювача (полоси):

$$L = A * n = 4 * 1,5 = 6 \text{ м. (при розташуванні заземлювачів по контуру).}$$

де n – ітераційна кількість вертикальних заземлювачів.

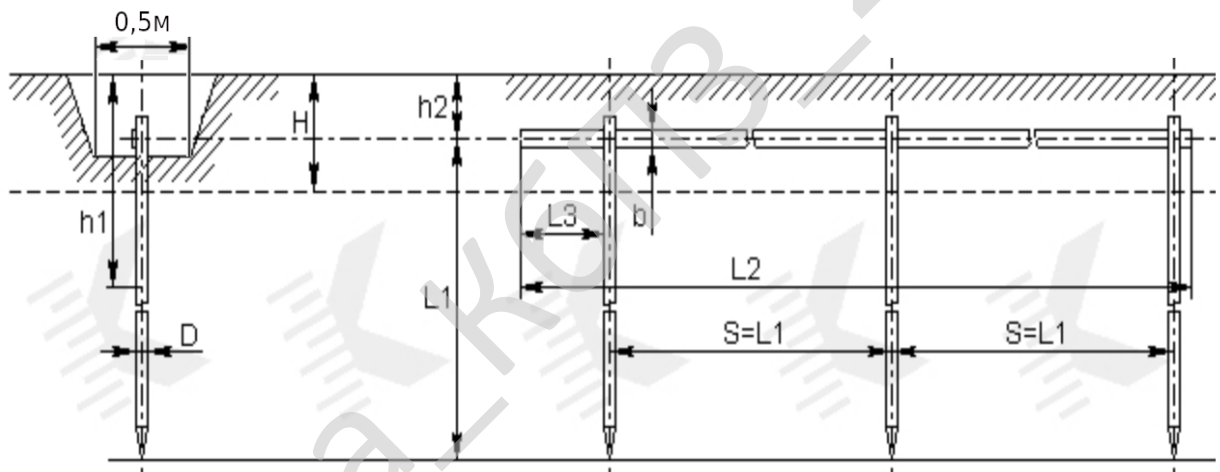


Рисунок 8.1 – Схема штучного робочого заземлення нульової точки трансформатора

Опір розтіканню вертикальних заземлювачів:

$$R_B = R_{\Pi} R / (R_{\Pi} - R) = 11,1 \text{ Ом.}$$

Кількість вертикальних заземлювачів:

$$N = R_0 / (R_B * \eta_B) = 26,28 / (11,1 * 0,69) = 3,4 \text{ шт.}$$

Остаточну кількість вертикальних заземлювачів приймаємо:

$$N = 4 \text{ шт.}$$

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-122.21.0094.00.00.ПЗ

Арк.

103

8.6 Висновки до розділу

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

З цих міркувань було здійснено аналіз приміщення, призначеного для праці програмістів, проведено розгляд небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи. Виконано розрахунок штучного освітлення, як одного з ключових факторів впливу на працездатність та здоров'я програміста. Розроблено заходи з охорони праці.

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		104

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи IP-телефонії з забезпеченням конфіденційності.

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 IP-телефонії з забезпеченням конфіденційності.

Рішення даного завдання полягало у вирішенні наступних задач:

–Був проведений огляд існуючих систем IP-телефонії з забезпеченням конфіденційності.

–Досліджена система IP-телефонії з забезпеченням конфіденційності.

–На основі отриманих результатів досліджень створена програмна реалізація системи IP-телефонії з забезпеченням конфіденційності.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання IP-телефонії з забезпеченням конфіденційності.

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-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		105

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

Програма реалізована на мові високого рівня RAD Studio Delphi 10. 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 XP/Vista/7/8/10.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм NTRU.

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 2754 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 2,6 роки.

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		106

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Піскова О.А. Дослідження та програмна реалізація системи IP-телефонії з забезпеченням конфіденційності // Збірник праць молодих науковців ЦНТУ. – Вип. 12. – Кропивницький: ЦНТУ, 2022.
2. Бакланов И.Г. ISDN и IP-телефония / Вестник связи, 1999, №4.
3. Будников В.Ю., Пономарев Б.А. Технологии обеспечения качества обслуживания в мультисервисных сетях / Вестник связи, 2000. №9.
4. Варламова Е. IP-телефония в России / Connect! Мир связи, 1999, №9.
5. Гольдштейн Б.С. Сигнализация в сетях связи. Том 1. М.: Радио и связь, 1998.
6. Гольдштейн Б.С. Протоколы сети доступа. Том 2. М.: Радио и связь, 1999.
7. Гольдштейн Б.С., Ехриель И.М., Рерле Р.Д. Интеллектуальные сети. М.: Радио и связь, 2000.
8. Кузнецов А.Е., Пинчук А. В., Суховицкий А.Л. Построение сетей IP-телефонии / Компьютерная телефония, 2000, №6.
9. Кульгин М. Технологии корпоративных сетей. Изд. «Питер», 1999.
10. Ломакин Д. Технические решения IP-телефонии / Мобильные системы, 1999 №8.
11. Мюнх Б., Скворцова С. Сигнализация в сетях IP-телефонии. -Часть I, II/Сети и системы связи, 1999. – №13(47), 14(48).
12. ITU-T Recommendation G.723.1. Dual Rate speech coder for multimedia communication transmitting at 5.3 and 6.3 kbit / sec. – 1996.
13. ITU-T Recommendation G.729. Speech codec for multimedia telecommunications transmitting at 8 / 13 kbit / s. – 1996.
14. ITU-T Recommendation H.225.0. Call signaling protocols and media stream packetization for packet-based multimedia communication systems. -Geneva,

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		107

1998.

15. ITU-T Recommendation H.245. Control protocol for multimedia communication. -Geneva, 1998

16. ITU-T Recommendation H.248. Gateway control protocol. – Geneva, 2000.

17. ITU-T Recommendation H.320. Narrow-band Visual Telephone Systems and Terminal Equipment. – 1996.

18. ITU-T Recommendation H.321. Adaptation of H.320 Visual Telephone Terminals to B-ISDN Environments. – 1996.

19. ITU-T Recommendation H.322. Visual Telephone Systems and Terminal Equipment for Local Area Networks which Provide a Guaranteed Quality of Service. – 1996.

20. ITU-T Recommendation H.323. Packet based multimedia communication systems. – Geneva, 1998.

21. ITU-T Recommendation H.324. Terminal for Low Bit Rate Multimedia Communications. -1996.

22. ITU-T Recommendation Q.931. ISDN User-Network Interface Layer 3 Specification for Basic Call Control. – 1993.

23. RFC 2205. Resource Reservation Protocol (RSVP). Ver.1. Functional Specification. – September 1997.

24. RFC 2327. Session Description Protocol. M. Handley, V. Jacobson. April, 1998.

25. RFC 2543. SIP: Session Initiation Protocol. M. Handley, H. Schulzrinne, E. Schooler, J. Rosenberg. March 1999.

26. RFC 2705. Media Gateway Control Protocol (MGCP) Version 1.0. M. Arango, A. Dugan, I. Elliott, C. Huitema, S. Pickett. October 1999.

27. RFC 2865. Remote Authentication Dial In User Service (RADIUS). C.Rigney, S. Willens, A. Rubens, W. Simpson. June 2000.

28. RFC 2885. Megaco Protocol 0.8. A.Rayhan, B. Rosen, J. Segers. August

					БКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		108

2000.

29. Дреев А.Н. Использование неравномерного распределения единичных битов для дополнительного сжатия SPIHT кода / А.Н. Дреев, А.А. Смирнов // Информационные системы в управлении, образовании, промышленности: монография. Под редакцией профессора В.С. Пономаренко. – Х.: Вид-во ТОВ «Щедра садиба плюс», 2014. – С. 498.

30. Дреев О.М. Дослідження впливу шляху розгортки на ступінь ентропійного стиснення цифрового зображення / О.М. Дреев, О.В. Слюсар // Збірник наукових праць Кіровоградського національного технічного університету. Техніка в сільськогосподарському виробництві, галузеве машинобудування, автоматизація. Випуск 21. – Кіровоград: КНТУ. – 2008 – С. 115-118.

31. Дреев О.М. Метод розвантаження телекомунікаційного сервера за рахунок кешування зображень / О.М. Дреев // Збірник наукових праць Кіровоградського національного технічного університету. Техніка в сільськогосподарському виробництві, галузеве машинобудування, автоматизація. Випуск 25. Ч. I. – Кіровоград: КНТУ. – 2012 – С. 419-424.

32. Дреев О.М. Метод прогнозування завантаженості серверу телекомунікаційної мережі / О.М. Дреев, О.А. Смірнов, Є.В. Мелешко, О.В. Коваленко // Системи обробки інформації. Випуск 3(101) Том 2. – Х.: ХУПС. – 2012. – С. 181-188.

33. Дреев О.М. Оцінка якості стиснення зображень на основі дискретного перетворення Хартлі / О.В. Коваленко, О.П. Доренський, О.М. Дреев // Системи озброєння і військова техніка. Науковий журнал 2(34)– Х.: ХУПС – 2013. С. 99-102.

34. Дреев О.М. Дослідження впливу ступеня стиснення зображень на оперативність їх доставки у телекомунікаційній системі / О.А. Смірнов, О.М. Дреев, О.П. Доренський // Збірник наукових праць "Системи обробки інформації". – Випуск 8(115). – Х.: ХУПС – 2013. – С. 234-239.

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		109

35. Дреев А.Н. Сравнение битовых плотностей при использовании различных методов кодирования информации / А.Н. Дреев, А.А. Смирнов // Системи обробки інформації, 2014, випуск 2 (118), том 2– Харків: ХУПС – 2014. С 64-66.

36. Дреев О.М. Моделювання впливу інтенсивності трафіку на оперативність доставляння інформації / О.М. Дреев // Науково-виробничий журнал “Зв’язок”. – Київ: ДУТ, 2014. – № 2 (108) С. 24-29.

37. Дреев А.Н. Повышение вероятности доставки сообщений в телекоммуникационных системах и сетях для обеспечения информационной безопасности / А.Н. Дреев, А.А. Смирнов // «Безпека інформації» Том 21, №1 2015 р. – Київ: НАУ – 2015. – С. 22-28.

38. Дреев О.М. Узагальнення вейвлету Хаара / О.М. Дреев, Г.М. Дреева // Збірник тез доповідей Комбінаторні конфігурації та їх застосування, 15-16 жовтня 2010 р. – Кіровоград – С. 58

39. Дреев О.М. Узагальнення вейвлету Хаара / О.М. Дреев // Матеріали науково-практичної конференції, присвяченої 80-річчю фізико-математичного факультету КДПУ ім. В. Винниченка 26 листопада 2010 р. – Кіровоград – С. 12

40. Дреев О.М. Метод прогнозування завантаженості серверу телекомунікаційної мережі / О.М. Дреев, О.В. Коваленко // Тези доповідей Новітні технології – для захисту повітряного простору. Дев’ята наукова конференція. 18-19 квітня 2011 р. – Х.: ХУПС. – 2012. – С. 206

41. Дреев О.М. Метод довгострокового прогнозування навантаження серверу телекомунікаційної мережі / О.М. Дреев, Г.М. Дреева // Комбінаторні конфігурації та їх застосування. Кіровоград. 13-14 квітня 2012 р. – Кіровоград: “Ексклюзів-систем”. – 2012. – С. 50

42. Дреев О.М. Вдосконалення стиснення зображень SPIHT методу шляхом додаткового кодування та відкладеної передачі уточнення вейвлет коефіцієнтів / О.М. Дреев // Дискретна математика та її застосування у економіко-

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		110

математичному моделюванні та інформаційних технологіях. 11-13 жовтня 2012 р. – Запоріжжя: ЗНУ – 2012. – С. 22-23.

43. Дреєв О.М. Методи підвищення якості обслуговування у телекомунікаційних системах та мережах / О.М. Дреєв, Г.М. Дреєва, О.А. Смірнов // Збірник тез доповідей. Академія внутрішніх військ МВС України “Застосування інформаційних технологій у підготовці та діяльності сил охорони правопорядку” 20-21 березня 2013р. – Харків: АВВ. – 2013. С. – 18-19

44. Дреєв А.Н. SPIHT кодирование с отложенной передачей значимых битов / А.Н. Дреєв // Тези доповідей. Новітні технології – для захисту повітряного простору. Дев'ята наукова конференція 17 квітня 2013 р. – Х.: ХУПС. – 2013. – С. 206

45. Дреєв А.Н. Повышение оперативности доставки данных повышенной востребованности в телекоммуникационных системах и сетях / А.Н. Дреєв, А.А. Смирнов, Е.В. Мелешко // Проблеми і перспективи розвитку ІТ-індустрії 25-26 квітня 2013 р. Системи обробки інформації. – Випуск 3 (110). Том 2. – Харків: ХУПС. – 2013. С. – 199.

46. Дреєв О.М. Середньостатистичний та найімовірніший час доставки багатопакетного повідомлення в телекомунікаційній системі або мережі / О.М. Дреєв, О.А. Смірнов // V Всеукраїнська науково-практична конференція "Інформатика та системні науки" ІСН – 2014, 13-15 березня 2014 року, м. Полтава – С. 92

47. Дреєв О.М. Визначення оптимального розміру блоку при бітовому арифметичному кодуванні / О.М. Дреєв, Г.М. Дреєва // Збірник тез доповідей Комбінаторні конфігурації та їх застосування, 11-12 квітня 2014 р. – Кіровоград – С. 44

48. Дреєв А.Н. Экстраполяция квазипериодических процессов с аддитивными помехами / А.Н. Дреєв, А.А. Смирнов // П'ята Міжнародна науково-практична конференція "Інформаційні технології та моделювання в економіці" 15-16 травня 2014 р. – Черкаси – С. 59

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		111

49. Дреев А.Н. Статистическая модель передачи многопакетного сообщения в телекоммуникационной системе или сети / А.Н. Дреев, А.А. Смирнов // «Компьютерное моделирование в наукоемких технологиях (КМНТ-2014)» Харьков, 28-31 мая 2014 года – С. 137-140

50. ДСТУ 2481 – 94 Системи оброблення інформації інтелектуальні інформаційні технології. Терміни та визначення. – Х.: ДЕРЖСТАНДАРТ УКРАЇНИ, 1994. – 33 с.

51. ДСТУ В 3265 – 95. Зв'язок військовий. Терміни та визначення. – К.:УкрНДІССІ, 1995. – 23 с.

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

53. Про охорону праці: Закон України від 14.10.1992 р. № 2694-ХІІ. Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/2694-12#Text>

54. Сакулин В.П., Шептовицкий В.М. Безопасность труда при монтаже и эксплуатации электроустановок / В.П.Сакулин, В.М.Шептовицкий. – Л. : “Колос”, 1973. – 238 с.

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

					ВКРМ-122.21.0094.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		112

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-122.21.0094.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Піскова О.А.				Дослідження та програмна реалізація системи IP-телефонії з забезпеченням конфіденційності	Літ.	Аркуш	Аркушів
Перевірів	Смірнова Т.В.					М	1	6
Н. Контр.	Гермак В.С.				ЦНТУ КН-20МЗ			
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи IP-телефонії з забезпеченням конфіденційності.

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. № 40-13 від 02.08.2021 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи IP-телефонії з забезпеченням конфіденційності.

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-122.21.0094.00.00.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 IP-телефонії з забезпеченням конфіденційності;
- 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-122.21.0094.00.00.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 XP/Vista/7/8/10 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 XP/Vista/7/8/10.

5.8.1 Обладнання

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

5.8.2 Мова програмування

Середовище RAD Studio Delphi 10.

					ВКРМ-122.21.0094.00.00.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 2021 року.

8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинна бути розглянута розробка заходів з умов поліпшення охорони праці.

					ВКРМ-122.21.0094.00.00.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 – 112 аркушів.

10 Етапи розробки

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

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

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

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Розрахунок з техніко-економічного обґрунтування.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

11.1 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на попередній захист 10.12.2021 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 23.12.2021 р.

					ВКРМ-122.21.0094.00.00.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

_____ Смірнова Т.В.

*Дослідження та програмна реалізація
системи IP-телефонії з забезпеченням конфіденційності*

Лістинг програми

Код документу 12

Носій: CD/DVD-диск

Загальна кількість аркушів: 48

Літера: РП

Кропивницький – 2021 року

Програма організації зв'язку

```

program iptel;

uses
  Forms,
  Dialogs,
  Windows,
  Sysutils,
  Messages,
  usimple in 'usimple.pas' {Form1},
  Unit2 in 'Unit2.pas' {Form2},
  Unit3 in 'Unit3.pas' {Form3},
  Unit1 in 'Unit1.pas' {Form0},
  Unit4 in 'Unit4.pas' {Form4};
{$R simple.RES}
var
  i:integer;
{
  function Crypt(Text,Key: String; Encode: boolean): String;
  var
    i, KeyLength: integer;
    Sign: ShortInt;
  begin
    KeyLength:=Length(Key);
    if Encode then Sign :=-1 else Sign:=1;
    for i:=1 to Length(Text) do
      Text[i]:=chr(ord(Text[i])+Sign*ord(Key[i mod KeyLength+1]));
    Result:=Text;
  end;}
begin
Application.Initialize;
{
if FileExists('main.dat')=false then
begin
  MessageDlg('Файл main.dat не знайдено! завершення
програми',mtInformation,[mbOK],0);
  Application.Terminate;
end else
begin
  AssignFile(F, 'main.dat');
  Reset(F);
  Readln(F, S);
  CloseFile(F);
  if Crypt(InputBox('Увага!', 'Введіть пароль',Y),KEY,false)=S then
  begin
    MessageDlg('Дякую, пароль вірний!', mtInformation,[mbOK],0);}

```

```
Try
    Form2:=TForm2.Create(Application);
    Form2.Show;
    Form2.Update;
    for i:=1 to 10 do
    begin
        sleep(200);
        Form2.ProgressBar1.Position:=i*10;
        Form2.Update;
    end;
    Application.Title := 'IpTel';
Application.CreateForm(TForm1, Form1);
Application.CreateForm(TForm3, Form3);
Application.CreateForm(TForm0, Form0);
Application.CreateForm(TForm4, Form4);
Finally
    Form2.Free;
End;
Application.Run;
{
    end
    else
    begin
        MessageDlg('Невірний пароль!', mtInformation, [mbOK], 0);
        Application.Terminate;
    end;
end;}
end.
```

```
unit U_SPLASH;

interface

uses

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

type
    TForm_SPLASH = class(TForm)
        Image1: TImage;
        Label1: TLabel;
        Gauge1: TGauge;
        Timer1: TTimer;
    private
        { Private declarations }
    public
        { Public declarations }
    end;

var
    Form_SPLASH: TForm_SPLASH;

implementation

{$R *.dfm}

end.
```

Головне вікно програми

```
unit usimple;

interface

uses
    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
    telefoon, StdCtrls, ComCtrls, ExtCtrls, jpeg, Menus, Buttons;

type
    TForm1 = class(TForm)
        Telefoon1: TTelefoon;
        StatusBar1: TStatusBar;
        PageControl1: TPageControl;
        TabSheet1: TTabSheet;
        Panel1: TPanel;
        Panel3: TPanel;
        Panel4: TPanel;
        Panel2: TPanel;
        Label1: TLabel;
        Label4: TLabel;
        Edit1: TEdit;
        Button2: TButton;
        Button1: TButton;
        Panel5: TPanel;
        TabSheet2: TTabSheet;
        Panel6: TPanel;
        Label2: TLabel;
        Panel7: TPanel;
        MainMenu1: TMainMenu;
        Panel8: TPanel;
        ListBox1: TListBox;
        BitBtn1: TBitBtn;
        BitBtn2: TBitBtn;
        ListBox2: TListBox;
        Panel9: TPanel;
        Label3: TLabel;
        Panel10: TPanel;
        BitBtn3: TBitBtn;
        BitBtn4: TBitBtn;
        N1: TMenuItem;
        IP1: TMenuItem;
        N5: TMenuItem;
        N6: TMenuItem;
        N7: TMenuItem;
        N8: TMenuItem;
    end;
end;
```



```
N9: TMenuItem;
N10: TMenuItem;
BitBtn5: TBitBtn;
Memol: TMemo;
TabSheet3: TTabSheet;
Panel11: TPanel;
Image3: TImage;
Label5: TLabel;
Label6: TLabel;
Label7: TLabel;
Label8: TLabel;
N11: TMenuItem;
N12: TMenuItem;
ListBox3: TListBox;
Label9: TLabel;
Label10: TLabel;
N2: TMenuItem;
PB1: TProgressBar;
PB2: TProgressBar;
T1: TTimer;
Image1: TImage;
Timer1: TTimer;
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure ListBox1Click(Sender: TObject);
procedure ListBox2Click(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure BitBtn3Click(Sender: TObject);
procedure BitBtn2Click(Sender: TObject);
procedure BitBtn4Click(Sender: TObject);
procedure N10Click(Sender: TObject);
procedure BitBtn5Click(Sender: TObject);
procedure N12Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure N3Click(Sender: TObject);
procedure N6Click(Sender: TObject);
procedure N2Click(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure T1Timer(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;
```

```

var
    Form1: TForm1;

implementation

uses Unit3, Unit2, Unit4, Unit1;

var
    KEY:string='2#$T%&(*qwrdas@@@#45';
    {$R *.DFM}
procedure TForm1.Button1Click(Sender: TObject);
begin
    Telefoon1.placecall(Edit1.text);
    ListBox3.items.add('D:'+DateToStr(now)+'T:'+TimeToStr(now)+'
IP:'+Edit1.text);
    T1.Enabled:=true;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin
    Telefoon1.calling:=false;
    T1.Enabled:=false;
    PB1.Position:=0;
    PB2.Position:=0;
end;
procedure TForm1.ListBox1Click(Sender: TObject);
begin
    Edit1.text:=ListBox1.Items.Strings[ListBox1.ItemIndex];
end;
procedure TForm1.ListBox2Click(Sender: TObject);
begin
    Label4.Caption:=ListBox2.Items.Strings[ListBox2.ItemIndex];
end;
procedure TForm1.BitBtn1Click(Sender: TObject);
var
    InputString: string;
begin
    InputString:= InputBox('Додавання ip адреси', 'Прошу', 'Введення IP не
здійснено');
    if (InputString<>'Введення IP не здійснено') then
    begin
        Listbox1.Items.add(InputString);
    end;
end;
procedure TForm1.BitBtn3Click(Sender: TObject);
var
    InputString: string;

```

```

begin
  InputString:= InputBox('Додавання легенди ip адреси', 'Прошу', 'Введення
легенди не здійснено');
  if (InputString<>'Введення легенди не здійснено') then
  begin
    Listbox2.Items.add(InputString);
  end;
end;
procedure TForm1.BitBtn2Click(Sender: TObject);
begin
  ListBox1.Items.Strings[ListBox1.ItemIndex]:= '';
  ListBox1.Items.Delete(ListBox1.ItemIndex);
end;

procedure TForm1.BitBtn4Click(Sender: TObject);
begin
  ListBox2.Items.Strings[ListBox1.ItemIndex]:= '';
  ListBox2.Items.Delete(ListBox1.ItemIndex);
end;
procedure TForm1.N10Click(Sender: TObject);
begin
  Application.Terminate;
end;
procedure TForm1.BitBtn5Click(Sender: TObject);
var
  i:integer;
  G:boolean;
begin
  Form3.ListBox1.Clear;
  g:=false;
  for i:=0 to (ListBox1.Items.Count - 1) do
  begin
    if ListBox1.Selected[i] then
    begin
      Form3.ListBox1.Items.add(ListBox1.Items.Strings[i]);
      g:=true;
    end;
    if g then Form3.show;
  end;
end;
procedure TForm1.N12Click(Sender: TObject);
var
  F: TextFile;
  H,S:string;
  OLD:string;

  function Crypt(Text,Key: String; Encode: boolean): String;

```

```

var
  i, KeyLength: integer;
  Sign: ShortInt;
begin
  KeyLength:=Length(Key);
  if Encode then Sign :=-1 else Sign:=1;
  for i:=1 to Length(Text) do
    Text[i]:=chr(ord(Text[i])+Sign*ord(Key[i mod KeyLength+1]));
  Result:=Text;
end;
begin
if FileExists('main.dat') then
begin
  AssignFile(F, 'main.dat');
  Reset(F);
  Readln(F, S);
  CloseFile(F);
if Crypt(InputBox('Увага!', 'Введіть старий пароль', ''), KEY, false)=S then
begin
  H:=Crypt(InputBox('Увага!', 'Введіть новий пароль', ''), KEY, false);
  if H<>' ' then
  begin
    MessageDlg('Дякую пароль змінено', mtInformation, [mbOK], 0);
    AssignFile(F, 'main.dat');
    Rewrite(F);
    Writeln(F, H);
    CloseFile(F);
  end else MessageDlg('Введіть значення!', mtInformation, [mbOK], 0);
end
else
begin
  MessageDlg('Файл main.dat не знайдено чи пароль невірний! завершення
програми', mtInformation, [mbOK], 0);
  Application.Terminate;
end;
end;
end;
procedure TForm1.FormCreate(Sender: TObject);
begin
  randomize;
  PB1.Position:=0;
  PB2.Position:=0;
  T1.Enabled:=false;
  Timer1.Enabled:=true;
if FileExists('MainLegend.dat')=false then
begin

```

```

    MessageDlg('Файл MainLegend.dat не найдено!',mtInformation,[mbOK],0);
end else
begin
    ListBox2.Items.LoadFromFile('MainLegend.dat');
end;
if FileExists('MainIP.dat')=false then
begin
    MessageDlg('Файл MainIP.dat не найдено!',mtInformation,[mbOK],0);
end else
begin
    ListBox1.Items.LoadFromFile('MainIP.dat');
end;
if FileExists('MainHISTORY.dat')=false then MessageDlg('Файл MainHISTORY.dat
не найдено!',mtInformation,[mbOK],0)
else ListBox3.Items.LoadFromFile('MainHISTORY.dat');
if FileExists('mainHelp.dat')=false then MessageDlg('Файл mainHelp.dat не
найджено!',mtInformation,[mbOK],0)
else Memo1.Lines.LoadFromFile('mainHelp.dat');
end;
procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
if FileExists('MainLegend.dat')=false then MessageDlg('Файл MainLegend.dat не
найджено!',mtInformation,[mbOK],0)
else ListBox2.Items.SaveToFile('MainLegend.dat');
if FileExists('MainIP.dat')=false then MessageDlg('Файл MainIP.dat не
найджено!',mtInformation,[mbOK],0)
else ListBox1.Items.SaveToFile('MainIP.dat');
if FileExists('MainHISTORY.dat')=false then MessageDlg('Файл MainHISTORY.dat
не найдено!',mtInformation,[mbOK],0)
else ListBox3.Items.SaveToFile('MainHISTORY.dat');
end;
procedure TForm1.N3Click(Sender: TObject);
begin
TabSheet2.Visible:=true;
end;
procedure TForm1.N6Click(Sender: TObject);
begin
Telefoon1.calling:=false
end;
procedure TForm1.N2Click(Sender: TObject);
begin
Form0.show;
end;
procedure TForm1.FormShow(Sender: TObject);
begin
PB1.Position:=0;

```

```
PB2.Position:=0;
end;
procedure TForm1.T1Timer(Sender: TObject);
var
i1,i2:integer;
begin
PB1.Position:=random(100);
PB2.Position:=PB1.Position;
end;
procedure TForm1.Timer1Timer(Sender: TObject);
begin
Form4.show;
Form1.hide;
Timer1.Enabled:=false;
end;
end.
```

Кафедра КБПЗ – 2021 рік

Підпрограма налагодження звукової карти AudioSettings

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  ComCtrls, StdCtrls, AMixer, MMSystem;
type
  TForm0 = class(TForm)
    ComboBox1: TComboBox;
    ComboBox2: TComboBox;
    TrackBar: TTrackBar;
    CheckBox: TCheckBox;
    Label1: TLabel;
    Label2: TLabel;
    Mixer: TAudioMixer;
    Label3: TLabel;
    Label4: TLabel;
    ComboBox3: TComboBox;
    LabelStereo: TLabel;
    Button1: TButton;
    procedure FormCreate(Sender: TObject);
    procedure ComboBox1Change(Sender: TObject);
    procedure ComboBox2Change(Sender: TObject);
    procedure MixerControlChange(Sender: TObject; MixerH, ID: Integer);
    procedure TrackBarChange(Sender: TObject);
    procedure CheckBoxClick(Sender: TObject);
    procedure ComboBox3Change(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
  private
    { Private declarations }
    Setting: Boolean;
  public
    { Public declarations }
  end;
var
  Form0: TForm0;
implementation
uses usimple;
{$R *.DFM}
procedure TForm0.FormCreate(Sender: TObject);
var A: Integer;
begin
  For A := 0 to Mixer.MixerCount - 1 do
    ComboBox3.Items.Add ('Mixer '+IntToStr(A));
  If (ComboBox3.Items.Count > 0) then

```

```

        ComboBox3.ItemIndex := 0;
        ComboBox3Change (Sender);
    end;
    procedure TForm0.ComboBox1Change(Sender: TObject);
    var A:Integer;
    begin
        ComboBox2.Items.Clear;
        ComboBox2.Items.Add
        (Mixer.Destinations[ComboBox1.ItemIndex].Data.szName);
        For A:=0 to Mixer.Destinations[ComboBox1.ItemIndex].Connections.Count-1 do
        ComboBox2.Items.Add(Mixer.Destinations[ComboBox1.ItemIndex].Connections[A].Dat
        a.szName);
        If ComboBox2.Items.Count>0 then
        begin
            ComboBox2.ItemIndex:=0;    ComboBox2Change (Self);
        end;
    end;
    procedure TForm0.ComboBox2Change(Sender: TObject);
    var L,R,M:Integer;
        VD,MD:Boolean;
        Stereo:Boolean;
        IsSelect:Boolean;
    begin
        Mixer.GetVolume (ComboBox1.ItemIndex,ComboBox2.ItemIndex-
    1,L,R,M,Stereo,VD,MD,IsSelect);
        Setting:=True;
        TrackBar.Visible:=not VD;
        Label1.Visible:=not VD;
        Label3.Visible:=VD;
        If TrackBar.Visible then
            TrackBar.Position:=L;
        CheckBox.Visible:=not MD;
        Label2.Visible:=not MD;
        Label4.Visible:=MD;
        If CheckBox.Visible then
            CheckBox.Checked:=M<>0;
        If (Stereo) then
            LabelStereo.Caption := '- stereo -'
        else
            LabelStereo.Caption := '- mono -';
        Setting:=False;
    end;
    procedure TForm0.MixerControlChange(Sender: TObject; MixerH, ID: Integer);
    begin
        ComboBox2Change (Self);
    end;

```



```

procedure TForm0.TrackBarChange(Sender: TObject);
begin
  If (not Setting) then
  begin
    Setting:=True;
    Mixer.SetVolume                (ComboBox1.ItemIndex,ComboBox2.ItemIndex-
1,TrackBar.Position,TrackBar.Position,Integer(CheckBox.Checked));
    Setting:=False;
  end;
end;
procedure TForm0.CheckBoxClick(Sender: TObject);
begin
  If not Setting then
  begin
    Setting:=True;
    Mixer.SetVolume                (ComboBox1.ItemIndex,ComboBox2.ItemIndex-
1,TrackBar.Position,TrackBar.Position,Integer(CheckBox.Checked));
    Setting:=False;
  end;
end;
procedure TForm0.ComboBox3Change(Sender: TObject);
var A:Integer;
begin
  If (ComboBox3.ItemIndex >= 0) AND (ComboBox3.ItemIndex < Mixer.MixerCount)
then
  Mixer.MixerId := ComboBox3.ItemIndex;
  ComboBox1.Items.Clear;
  If Mixer.MixerCount>0 then
  begin
    For A:=0 to Mixer.Destinations.Count-1 do
      ComboBox1.Items.Add (Mixer.Destinations[A].Data.szName);
    If ComboBox1.Items.Count>0 then
    begin
      ComboBox1.ItemIndex:=0;    ComboBox1Change (Self);
    end;
  end
  else
  begin
    ComboBox1.OnChange:=nil;    ComboBox2.OnChange:=nil;
    TrackBar.OnChange:=nil;    CheckBox.OnClick:=nil;
    MessageDlg ('No mixer present in the system !',mtError,[mbOK],0);
  end;
  Setting:=False;
end;
procedure TForm0.Button1Click(Sender: TObject);
begin

```

```
Form0.hide; Form1.show;  
end;  
procedure TForm0.FormClose(Sender: TObject; var Action: TCloseAction);  
begin  
Form0.hide; Form1.show;  
end;  
end.
```

Кафедра КБПЗ – 2021 рік

Підпрограма виклику вікна налагодження звукової карти AudioSettings

```
unit Unit2;

interface

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

type
  TForm2 = class(TForm)
    Image1: TImage;
    RichEdit1: TRichEdit;
    ProgressBar1: TProgressBar;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form2: TForm2;

implementation

{$R *.DFM}

end.
```

Підпрограма селективного зв'язку

```
unit Unit3;

interface

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

type
  TForm3 = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    Panel3: TPanel;
    ListBox1: TListBox;
    Animat1: TAnimate;
    Panel4: TPanel;
    BitBtn2: TBitBtn;
    BitBtn1: TBitBtn;
    BitBtn3: TBitBtn;
    Image1: TImage;
    procedure BitBtn2Click(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn3Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form3: TForm3;

implementation

uses usimple;

{$R *.DFM}

procedure TForm3.BitBtn2Click(Sender: TObject);
begin
  form3.hide;
end;

procedure TForm3.BitBtn1Click(Sender: TObject);
begin
```

```
Animatel.Active:=true;  
form1.telefoon1.placecall(form1.edit1.text);  
end;  
  
procedure TForm3.BitBtn3Click(Sender: TObject);  
begin  
Form1.Telefoon1.calling:=false;  
Animatel.Active:=false;  
end;  
  
end.
```

Кафедра _ КБПЗ _ 2021 рік

Підпрограма парольного захисту

```

unit Unit4;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Buttons, Mask, ExtCtrls;
type
  TForm4 = class(TForm)
    Panel1: TPanel;
    M: TMaskEdit;
    BitBtn1: TBitBtn;
    procedure BitBtn1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form4: TForm4;
implementation
uses usimple;
VAR
  KEY:string='2%#$T%&(*qwrdas@@@#45';
  DAT:string='VUW';
{$R *.DFM}
  function Crypt(Text,Key: String; Encode: boolean): String;
  var
    i, KeyLength: integer;
    Sign: ShortInt;
  begin
    KeyLength:=Length(Key);
    if Encode then Sign :=-1 else Sign:=1;
    for i:=1 to Length(Text) do
      Text[i]:=chr(ord(Text[i])+Sign*ord(Key[i mod KeyLength+1]));
    Result:=Text;
  end;
  procedure TForm4.BitBtn1Click(Sender: TObject);
  var
    F: TextFile;
    i:integer;
    s:string;
    Y:string;
    UUU:string;
  begin
    if FileExists('main.dat')=TRUE then
      begin

```

```
AssignFile(F, 'main.dat');
Reset(F);
Readln(F, S);
CloseFile(F);
UUU:=Crypt(M.text,KEY,false);
if UUU=S then
    begin
        Form4.hide;
        Form1.show;
        MessageDlg('Дякую, пароль вірний!',mtInformation,[mbOK],0);
    end
    else MessageDlg('Введіть пароль!',mtInformation,[mbOK],0);
end
else begin
    MessageDlg('Файл main.dat не знайдено! завершення
програми',mtInformation,[mbOK],0);
    Application.Terminate;
end;
end;
end.
```

Кафедра КБПЗ – 2021 рік

Розроблений компонент налагодження звука

```

unit AMixer;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  MMSystem;

type
  TAudioMixer=class;

  TPListFreeItemNotify=procedure (Pntr:Pointer) of object;
  TMixerChange=procedure (Sender:TObject;MixerH:HMixer;ID:Integer) of object;
    {MixerH is handle of mixer, which sent this message.
     ID is ID of changed item (line or control).}

  TPointerList=class(TObject)
  private
    FOnFreeItem:TPListFreeItemNotify;
    Items:Tlist;
  protected
    function GetPointer (Ind:Integer):Pointer;
    function GetCount :integer;
  public
    constructor Create;
    destructor Destroy; override;
    procedure Clear;
    procedure Add (Pntr:Pointer);
    property Count:Integer read GetCount;
    property Pointer[Ind:Integer]:Pointer read GetPointer; default;
    property OnFreeItem:TPListFreeItemNotify read FOnFreeItem write
FOnFreeItem;
  end;

  TMixerControls=class(TObject)
  private
    heap:pointer;
    FControls:TPointerList;
  protected
    function GetControl (Ind:Integer):PMixerControl;
    function GetCount:Integer;
  public
    constructor Create (AMixer:TAudioMixer; AData:TMixerLine);
    destructor Destroy; override;
    property Control[Ind:Integer]:PMixerControl read GetControl; default;

```



```

    property Count:Integer read GetCount;
end;

TMixerConnection=class(TObject)
private
    XMixer:TAudioMixer;
    FData:TMixerLine;
    FControls:TMixerControls;
public
    constructor Create (AMixer:TAudioMixer; AData:TMixerLine);
    destructor Destroy; override;
    property Controls:TMixerControls read FControls;
    property Data:TMixerLine read FData;
end;

TMixerConnections=class(TObject)
private
    XMixer:TAudioMixer;
    FConnections:TPointerList;
protected
    procedure DoFreeItem (Pntr:Pointer);
    function GetConnection (Ind:Integer):TMixerConnection;
    function GetCount:Integer;
public
    constructor Create (AMixer:TAudioMixer; AData:TMixerLine);
    destructor Destroy; override;
    property Connection[Ind:Integer]:TMixerConnection read GetConnection;
default;
    property Count:Integer read GetCount;
end;

TMixerDestination=class(TObject)
private
    XMixer:TAudioMixer;
    FData:TMixerLine;
    FControls:TMixerControls;
    FConnections:TMixerConnections;
public
    constructor Create (AMixer:TAudioMixer; AData:TMixerLine);
    destructor Destroy; override;
    property Connections:TMixerConnections read FConnections;
    property Controls:TMixerControls read FControls;
    property Data:TMixerLine read FData;
end;

TMixerDestinations=class(TObject)

```

```

private
    FDestinations:TPointerList;
protected
    function GetDestination (Ind:Integer):TMixerDestination;
    procedure DoFreeItem (Pntr:Pointer);
    function GetCount:Integer;
public
    constructor Create (AMixer:TAudioMixer);
    destructor Destroy; override;
    property Count:Integer read GetCount;
    property Destination[Ind:Integer]:TMixerDestination read GetDestination;
default;
end;

TAudioMixer = class(TComponent)
private
    XWndHandle:HWND;

    FDestinations:TMixerDestinations;
    FMixersCount:Integer;
    FMixerHandle:HMixer;
    FMixerId:Integer;
    FMixerCaps:TMixerCaps;
    FDriverVersion: MMVERSION;
    FManufacturer: String;
    FProductId: Word;
    FNumberOfLine: Integer;
    FProductName: String;
    FOnLineChange:TMixerChange;
    FOnControlChange:TMixerChange;
protected
    procedure SetMixerId (Value:Integer);
    procedure MixerCallBack (var Msg:TMessage);
    procedure CloseMixer;
published
    constructor Create (AOwner:TComponent); override;
    destructor Destroy; override;
    property DriverVersion: MMVERSION read FDriverVersion;
    property ProductId: WORD read FProductId;
    property NumberOfLine: Integer read FNumberOfLine;
    property Manufacturer: string read FManufacturer;
    property ProductName: string read FProductName;
    property MixerId:Integer read FMixerId write SetMixerId;
    {Opened mixer - value must be in range 0..MixersCount-1
     If no mixer is opened this value is -1}
    property OnLineChange:TMixerChange read FOnLineChange write FOnLineChange;

```

```

property OnControlChange:TMixerChange read FOnControlChange write
FOnControlChange;
public
function GetVolume (ADestination, AConnection:Integer; var LeftVol,
RightVol, Mute:Integer; var Stereo, VolDisabled, MuteDisabled,
MuteIsSelect:Boolean):Boolean;
    {This function return volume of selected Destination and Connection.
    ADestination must be from range 0..Destinations.Count-1
    AConnection must be in range
    0..Destinations[ADestination].Connections.Count-1
    If you want to read master volume of some Destination, you have to
    set AConnection to -1.
    If LeftVol, RightVol or Mute is not supported by queried connection,
    it's return value will be -1.

    LeftVol and RightVol are in range 0..65536

    If Mute is non-zero then the connection is silent (or vice-versa - see
MuteIsSelect parameter)
    If specified line is recording source then Mute specifies if programs
will
    record from this connection (it is copy of "Select" Checkbox in
    standard Windows Volume Control program)
    Stereo is true, then this control is stereo.
    VolDisabled or MuteDisabled is True when you cannot apply settings to
this
    control (but can read it).
    MuteIsSelect returns True is "mute" work here as select - opposite of
mute.

    Return value of the function is True if no error has ocured,
    otherwise it returns False.}
function SetVolume (ADestination, AConnection:Integer; LeftVol, RightVol,
Mute:Integer):Boolean;
    {This function sets volume.
    If you set RightVol to -1 and connection is stereo then LeftVol will be
    copied to RightVol.
    If LeftVol or Mute is -1 then this value will not be set.
    Note that "Mute" can be "select" (which is reversed mute) - see
function
    GetVolume, parameter MuteIsSelect.

    Return value is True if ADestination and AConnection are correct,
    otherwise False.}

```

```

function GetPeak(ADestination, AConnection:Integer; var LeftPeak,
RightPeak:Integer):Boolean;

function GetMute (ADestination, AConnection:Integer; var
Mute:Boolean):Boolean;

function SetMute (ADestination, AConnection:Integer; Mute:Boolean):Boolean;

property Destinations:TMixerDestinations read FDestinations;
  {Ind must be in range 0..DestinationsCount-1}
property MixerCaps:TMixerCaps read FMixerCaps;
property MixerCount:Integer read FMixersCount;
  {Number of mixers present in system; mostly 1}
property MixerHandle:HMixer read FMixerHandle;
  {Handle of opened mixer}
end;

procedure Register;

implementation

{-----}
{TPointerList}
{-----}

constructor TPointerList.Create;
begin
  Items := TList.Create;
end;

destructor TPointerList.Destroy;
begin
  Clear;
  Items.Free;
end;

procedure TPointerList.Add (Ptr:Pointer);
begin
  Items.Add (Ptr);
end;

function TPointerList.GetPointer (Ind:Integer):Pointer;
begin
  Result := nil;
  If (Ind < Count) then
    Result := Items[Ind];
end;

```

```

procedure TPointerList.Clear;
var I:Integer;
begin
  for I := 0 to Items.Count-1 do begin
    If Assigned (FOnFreeItem) then
      FOnFreeItem (Items[I])
    end;
    Items.Clear;
  end;

function TPointerList.GetCount:Integer;
begin
  Result := Items.Count;
end;

{-----}
{TMixerControls}
{-----}
constructor TMixerControls.Create (AMixer:TAudioMixer; AData:TMixerLine);
var MLC:TMixerLineControls;
    A,B:Integer;
    P:PMixerControl;
begin
  FControls := TPointerList.Create;
  GetMem (P, SizeOf(TMixerControl)*AData.cControls);
  heap := P;
  MLC.cbStruct := SizeOf(MLC);
  MLC.dwLineID := AData.dwLineID;
  MLC.cbmxctrl := SizeOf(TMixerControl);
  MLC.cControls := AData.cControls;
  MLC.pamxctrl := P;
  A := MixerGetLineControls(AMixer.MixerHandle, @MLC,
MIXER_GETLINECONTROLSF_ALL);
  If A = MMSYSERR_NOERROR then
  begin
    For B := 0 to AData.cControls-1 do
    begin
      FControls.Add (P);
      P := PMixerControl (DWORD(P) + sizeof (TMixerControl));
    end;
  end;
end;

destructor TMixerControls.Destroy;
begin
  FControls.free;

```

```

    freemem(heap);
    inherited;
end;

function TMixerControls.GetControl (Ind:Integer):PMixerControl;
begin
    Result := FControls.Pointer[Ind];
end;

function TMixerControls.GetCount:Integer;
begin
    Result := FControls.Count;
end;

{-----}
{TMixerConnection}
{-----}

constructor TMixerConnection.Create (AMixer:TAudioMixer; AData:TMixerLine);
begin
    FData := AData;
    XMixer := AMixer;
    FControls := TMixerControls.Create (AMixer, AData);
end;

destructor TMixerConnection.Destroy;
begin
    FControls.Free;
    inherited;
end;

{-----}
{TMixerConnections}
{-----}

constructor TMixerConnections.Create (AMixer:TAudioMixer; AData:TMixerLine);
var A,B:Integer;
    ML:TMixerLine;
begin
    XMixer := AMixer;
    FConnections := TPointerList.Create;
    FConnections.OnFreeItem := Dofreeitem;
    ML.cbStruct := SizeOf(TMixerLine);
    ML.dwDestination := AData.dwDestination;
    For A := 0 to AData.cConnections-1 do
        begin

```

```

    ML.dwSource := A;
    B := MixerGetLineInfo (AMixer.MixerHandle, @ML,
MIXER_GETLINEINFOF_SOURCE);
    If B = MMSYSERR_NOERROR then
        FConnections.Add (Pointer(TMixerConnection.Create (XMixer, ML)));
    end;
end;

destructor TMixerConnections.Destroy;
begin
    FConnections.Free;
    inherited;
end;

procedure TMixerConnections.DoFreeItem (Pntr:Pointer);
begin
    TMixerConnection(Pntr).Free;
end;

function TMixerConnections.GetConnection (Ind:Integer):TMixerConnection;
begin
    Result := FConnections.Pointer[Ind];
end;

function TMixerConnections.GetCount:Integer;
begin
    Result := FConnections.Count;
end;

{-----}
{TMixerDestination}
{-----}

constructor TMixerDestination.Create (AMixer:TAudioMixer; AData:TMixerLine);
begin
    FData := AData;
    XMixer := AMixer;
    FConnections := TMixerConnections.Create (XMixer, FData);
    FControls := TMixerControls.Create (XMixer, AData);
end;

destructor TMixerDestination.Destroy;
begin
    Fcontrols.Free;
    FConnections.Free;
    inherited;
end;

```

```

end;

{-----}
{TMixerDestinations}
{-----}

constructor TMixerDestinations.Create (AMixer:TAudioMixer);
var A,B:Integer;
    ML:TMixerLine;
begin
    FDestinations := TPointerList.Create;
    FDestinations.OnFreeItem := DoFreeItem;
    if (AMixer = nil) then
        Exit;
    For A := 0 to AMixer.MixerCaps.cDestinations-1 do
    begin
        ML.cbStruct := SizeOf(TMixerLine);
        ML.dwDestination := A;
        B := MixerGetLineInfo (AMixer.MixerHandle, @ML,
MIXER_GETLINEINFOF_DESTINATION);
        If B = MMSYSERR_NOERROR then
            FDestinations.Add (Pointer(TMixerDestination.Create (AMixer, ML)));
        end;
    end;
end;

procedure TMixerDestinations.DoFreeItem (Pntr:Pointer);
begin
    TMixerDestination(Pntr).Free;
end;

destructor TMixerDestinations.Destroy;
begin
    FDestinations.Free;
    inherited;
end;

function TMixerDestinations.GetDestination (Ind:Integer):TMixerDestination;
begin
    Result := nil;
    If (Assigned (FDestinations)) then
        Result := FDestinations.Pointer[Ind];
    end;
end;

function TMixerDestinations.GetCount:Integer;
begin
    Result := FDestinations.Count;
end;

```



```

end;

{-----}
{TAudioMixer}
{-----}

constructor TAudioMixer.Create (AOwner:TComponent);
begin
  inherited Create (AOwner);
  XWndHandle := AllocateHwnd (MixerCallBack);
  FMixersCount := mixerGetNumDevs;
  FMixerId := -1;
  if (FMixersCount = 0) then
    FDestinations := TMixerDestinations.Create (nil)
  else
    begin
      FDestinations := nil;
      SetMixerId (0);
    end;
end;

destructor TAudioMixer.Destroy;
begin
  CloseMixer;
  if XWndHandle <> 0 then
    DeAllocateHwnd (XWndHandle);
  inherited;
end;

procedure TAudioMixer.CloseMixer;
begin
  If FMixerId >= 0 then
    begin
      mixerClose (FMixerHandle);
      FMixerId := -1;
    end;
  FDestinations.Free;
  FDestinations := nil;
end;

procedure TAudioMixer.SetMixerId (Value:Integer);
label AllOK;
begin
  If (Value < 0) OR (Value >= FMixersCount) then
    Exit;
  CloseMixer;

```

```

    If mixerOpen (@FMixerHandle, Value, XWndHandle, 0, CALLBACK_WINDOW OR
MIXER_OBJECTF_MIXER) = MMSYSERR_NOERROR then
        goto ALLOK;

    // we will go here very rarely, but sometimes it could help
    If mixerOpen (@FMixerHandle, Value, XWndHandle, 0, CALLBACK_WINDOW) =
MMSYSERR_NOERROR then
        goto ALLOK;
    If mixerOpen (@FMixerHandle, Value, 0, 0, 0) = MMSYSERR_NOERROR then
        goto ALLOK;

    // an error has occurred
    FMixerId := -1;
    FDestinations := TMixerDestinations.Create (nil);

    Exit;

ALLOK:
    FMixerId := Value;
    mixerGetDevCaps (MixerId, @FMixerCaps, SizeOf (TMixerCaps));

    if FMixerCaps.wMid = MM_MICROSOFT then
        FManufacturer := 'Microsoft'
    else
        FManufacturer := IntToStr(FMixerCaps.wMid) + ' = Unknown';
    FDriverVersion := FMixerCaps.vDriverVersion;
    FProductId := FMixerCaps.wPid;
    FProductName := StrPas(FMixerCaps.szPName);
    FNumberOfLine := FMixerCaps.cDestinations;

    FDestinations := TMixerDestinations.Create (Self);
end;

procedure TAudioMixer.MixerCallBack (var Msg:TMessage);
begin
    case Msg.Msg of
        MM_MIXM_LINE_CHANGE:
            If Assigned (OnLineChange) then
                OnLineChange (Self, Msg.wParam, Msg.lParam);
        MM_MIXM_CONTROL_CHANGE:
            If Assigned (OnControlChange) then
                OnControlChange (Self, Msg.wParam, Msg.lParam);
    else
        Msg.Result := DefWindowProc (XWndHandle, Msg.Msg, Msg.WParam,
Msg.LParam);
    end;
end;

```

```

    end;
end;

const MIXER_LONG_NAME_CHARS = 64;

type MIXERCONTROLDETAILS_LISTTEXT = record
    dwParam1:DWORD;
    dwParam2:DWORD;
    szName:Array [0..MIXER_LONG_NAME_CHARS-1] of Char;
end;

type ListTextArray = array [0..1000] of MIXERCONTROLDETAILS_LISTTEXT;

function TAudioMixer.GetVolume (ADestination,AConnection:Integer;var LeftVol,
    RightVol, Mute:Integer;var Stereo, VolDisabled, MuteDisabled,
    MuteIsSelect:Boolean):Boolean;
var MD:TMixerDestination;
    MC:TMixerConnection;
    Cntrl:TMixerControls;
    MCD:TMixerControlDetails;
    Cntrl:PMixerControl;
    A,B:Integer;
    ML:TMixerLine;
    details:array [0..100] of Integer;
    ltext:^ListTextArray;
    MCDText:TMixerControlDetails;

begin
    Result := False;
    If (not Assigned (EDestinations)) then
        Exit;
    Stereo := False;
    MuteDisabled := True;
    VolDisabled := True;
    LeftVol := -1;
    RightVol := -1;
    Mute := -1;
    MuteIsSelect := False;
    MD := Destinations[ADestination];
    MC := nil;
    If AConnection <> -1 then
        MC := MD.Connections[AConnection];

    If MD <> nil then
        begin

```

```

Result := True;

//      If Mute = -1 then
begin
  If AConnection <> -1 then
  begin
    Cntrls := MD.Controls;
    ML := MD.Data;
    If (MC <> nil) AND (Cntrls <> nil) then
    begin
      A := 0;
      while (Mute = -1) AND (A < Cntrls.Count) do
      begin
        Cntrl := Cntrls[A];
        //          If (Cntrl.dwControlType AND MIXERCONTROL_CONTROLTYPE_MIXER =
MIXERCONTROL_CONTROLTYPE_MIXER) OR
//          (Cntrl.dwControlType AND MIXERCONTROL_CONTROLTYPE_MUX =
MIXERCONTROL_CONTROLTYPE_MUX) then

//          If (Cntrl.dwControlType AND MIXERCONTROL_CT_CLASS_MASK =
MIXERCONTROL_CT_CLASS_LIST) then

          If (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUX) then
            // Mux is similar to mixer, but only one line can be selected
at a time
          begin
            MCD.cbStruct := SizeOf(TMixerControlDetails);
            MCD.dwControlID := Cntrl.dwControlID;
            If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_UNIFORM > 0 then
              MCD.cChannels := 1
            else
              MCD.cChannels := ML.cChannels;
            If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_MULTIPLE =
MIXERCONTROL_CONTROLF_MULTIPLE then
              MCD.cMultipleItems := Cntrl.cMultipleItems
            else
              MCD.cMultipleItems := 0;
            MCD.cbDetails := 4;
            MCD.paDetails := @Details;
            B := mixerGetControlDetails
(FMixerHandle,@MCD,MIXER_GETCONTROLDETAILSF_VALUE);
            If B <> MMSYSERR_NOERROR then
            begin
              Inc (A);
              continue;
            end;

```

```

MCDText.cbStruct := sizeof (MCDText);
MCDText.dwControlID := Cntrl.dwControlID;
If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_UNIFORM > 0 then
  MCDText.cChannels := 1
else
  MCDText.cChannels := ML.cChannels;
  If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_MULTIPLE =
MIXERCONTROL_CONTROLF_MULTIPLE then
    MCDText.cMultipleItems := Cntrl.cMultipleItems
  else
    MCDText.cMultipleItems := 0;
  GetMem (ltext, MCDText.cChannels * MCDText.cMultipleItems *
sizeof (MIXERCONTROLDETAILS_LISTTEXT));
  MCDText.cbDetails := sizeof (MIXERCONTROLDETAILS_LISTTEXT);
  MCDText.paDetails := ltext;
  B := mixerGetControlDetails (FMixerHandle, @MCDText,
MIXER_GETCONTROLDETAILSF_LISTTEXT);
  If B <> MMSYSERR_NOERROR then
  begin
    FreeMem (ltext);
    Inc (A);
    continue;
  end;
  B := MCD.cChannels - 1;
  while (B < integer(MCD.cMultipleItems)) do
  begin
    if (ltext[B].dwParam1 = MC.Data.dwLineID) then
      break;
    Inc (B, MCD.cChannels);
  end;
  FreeMem (ltext);
  If (B < integer (MCD.cMultipleItems)) then
  begin
    Mute := Details[B];
    MuteDisabled := Cntrl.fdwControl AND
MIXERCONTROL_CONTROLF_DISABLED > 0;
    MuteIsSelect := True;
    break;
  end;
  end;
  Inc (A);
end;
end;
end;
end;
end;

```

```

end;

If AConnection = -1 then
begin
  Cntrls := MD.Controls;
  ML := MD.Data;
end
else
begin
  If MC <> nil then
  begin
    Cntrls := MC.Controls;
    ML := MC.Data;
  end
  else
    Cntrls := nil;
  end;
If Cntrls <> nil then
begin
  A := 0;
  while ((LeftVol = -1) OR (Mute = -1)) AND (A < Cntrls.Count) do
  begin
    Cntrl := Cntrls[A];
    If Cntrl <> nil then
    begin
      If ((Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_VOLUME) OR
          ((Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUTE) AND (Mute
= -1))) AND
          (Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_MULTIPLE <>
MIXERCONTROL_CONTROLF_MULTIPLE)
      then
      begin
        if (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUTE) then
          MCD.cbStruct := SizeOf(TMixerControlDetails)
        else
          MCD.cbStruct := SizeOf(TMixerControlDetails);
        MCD.dwControlID := Cntrl.dwControlID;
        If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_UNIFORM > 0 then
          MCD.cChannels := 1
        else
          MCD.cChannels := ML.cChannels;
        MCD.cMultipleItems := 0;
        MCD.cbDetails := SizeOf(Integer);
        MCD.paDetails := @details;
        B := mixerGetControlDetails (FMixerHandle, @MCD,
MIXER_GETCONTROLDETAILSF_VALUE);

```

```

    If B = MMSYSERR_NOERROR then
    begin
        If (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_VOLUME) AND
(LeftVol = -1) then
        begin
            VolDisabled          :=          Cntrl.fdwControl          AND
MIXERCONTROL_CONTROLF_DISABLED > 0;
            LeftVol := details[0];
            If MCD.cChannels > 1 then
            begin
                RightVol := Details[1];
                Stereo := True;
            end
            else
                RightVol := LeftVol;
            end
            else If (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUTE)
AND (Mute = -1) then
            begin
                MuteDisabled          :=          Cntrl.fdwControl          AND
MIXERCONTROL_CONTROLF_DISABLED > 0;
                If Details[0] <> 0 then
                    Mute := 1
                else
                    Mute := 0;
                end
            // NEW ->
            (*          else If (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_ONOFF)
AND (Mute = -1) then
            begin
                MuteDisabled          :=          Cntrl.fdwControl          AND
MIXERCONTROL_CONTROLF_DISABLED > 0;
                If Details[0] <> 0 then
                    Mute := 1
                else
                    Mute := 0;
                    MuteIsSelect := True;
                end;*)
            // <- NEW
            end;
        end;
        end;
        end;
        Inc (A);
    end;

{      If LeftVol = -1 then

```

```

        VolDisabled := True;
    If Mute = -1 then
        MuteDisabled := True;}
    end;
end;
end;

```

```

function TAudioMixer.SetVolume (ADestination, AConnection:Integer; LeftVol,
RightVol, Mute:Integer):Boolean;
var MD:TMixerDestination;
    MC:TMixerConnection;
    Cntrls:TMixerControls;
    MCD:TMixerControlDetails;
    Cntrl:PMixerControl;
    A,B:Integer;
    ML:TMixerLine;
    details:array [0..100] of Integer;
    VolSet,MuteSet:Boolean;
    ltext:^ListTextArray;
    MCDText:TMixerControlDetails;
begin
    Result := False;
    If (not Assigned (FDestinations)) then
        Exit;
    MC := nil;
    MD := Destinations[ADestination];
    If MD <> nil then
        begin
            If AConnection <> -1 then
                MC := MD.Connections[AConnection];

                VolSet := LeftVol = -1;
                MuteSet := Mute = -1;
                Result := True;

            If not MuteSet then
                begin
                    If AConnection <> -1 then
                        begin
                            Cntrls := MD.Controls;
                            ML := MD.Data;
                            If (MC <> nil) AND (Cntrls <> nil) then
                                begin
                                    A := 0;
                                    while not MuteSet AND (A < Cntrls.Count) do
                                        begin

```



```

Cntrl := Cntrls[A];
If (*(Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MIXER) OR*)
    (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUX) then
begin
    MCD.cbStruct := SizeOf(TMixerControlDetails);
    MCD.dwControlID := Cntrl.dwControlID;
    If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_UNIFORM > 0 then
        MCD.cChannels := 1
    else
        MCD.cChannels := ML.cChannels;
    If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_MULTIPLE =
MIXERCONTROL_CONTROLF_MULTIPLE then
        MCD.cMultipleItems := Cntrl.cMultipleItems
    else
        MCD.cMultipleItems := 0;
    MCD.cbDetails := 4;
    MCD.paDetails := @Details;
    MuteSet := True;
    mixerGetControlDetails (FMixerHandle, @MCD,
MIXER_GETCONTROLDETAILSF_VALUE);
    if (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUX) then
        For B := 0 to Cntrl.cMultipleItems-1 do
            Details[B] := 0;

            GetMem (ltext, MCD.cChannels * MCD.cMultipleItems * sizeof
(MIXERCONTROLDETAILS_LISTTEXT));
            MCDText.cbStruct := sizeof (MCDText);
            MCDText.dwControlID := Cntrl.dwControlID;
            MCDText.cChannels := MCD.cChannels;
            MCDText.cMultipleItems := MCD.cMultipleItems;
            MCDText.cbDetails := sizeof (MIXERCONTROLDETAILS_LISTTEXT);
            MCDText.paDetails := ltext;
            mixerGetControlDetails (FMixerHandle, @MCDText,
MIXER_GETCONTROLDETAILSF_LISTTEXT);
            B := MCD.cChannels - 1;
            while (B < integer (MCD.cMultipleItems)) do
            begin
                if (ltext[B].dwParam1 = MC.Data.dwLineID) then
                    break;
                Inc (B, MCD.cChannels);
            end;
            FreeMem (ltext);

            If (B < integer (MCD.cMultipleItems)) then
            begin
                Details[B] := Mute;

```

```

        mixerSetControlDetails          (FMixerHandle,          @MCD,
MIXER_GETCONTROLDETAILSF_VALUE);
        break;
    end;
end;
    Inc (A);
end;
end;
end;
end;
end;

If AConnection = -1 then
begin
    Cntrls := MD.Controls;
    ML := MD.Data;
end
else
begin
    If MC <> nil then
begin
    Cntrls := MC.Controls;
    ML := MC.Data;
end
else
    Cntrls := nil;
end;
If Cntrls <> nil then
begin

    A := 0;
    while (not VolSet OR not MuteSet) AND (A < Cntrls.Count) do
begin
    Cntrl := Cntrls[A];
    If Cntrl <> nil then
begin
        If (((Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_VOLUME) AND not
VolSet) OR
            ((Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUTE) AND not
MuteSet) (* NEW -> *) (*OR
            ((Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_ONOFF) AND not
MuteSet)*) (* <- NEW *) AND
            (Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_MULTIPLE <>
MIXERCONTROL_CONTROLF_MULTIPLE)
        then

```

```

begin
  MCD.cbStruct := SizeOf(TMixerControlDetails);
  MCD.dwControlID := Cntrl.dwControlID;
  If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_UNIFORM > 0 then
    MCD.cChannels := 1
  else
    MCD.cChannels := ML.cChannels;
  MCD.cMultipleItems := 0;
  MCD.cbDetails := SizeOf(Integer);
  MCD.paDetails := @Details;
  If (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_VOLUME) then
    begin
      Details[0] := LeftVol;
      If RightVol = -1 then
        Details[1] := LeftVol
      else
        Details[1] := RightVol;
      VolSet := True;
    end
  else if ((Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUTE) (*
NEW -> *) (* OR
(Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_ONOFF) *)
(* <- NEW *) ) then
    begin
      For B := 0 to MCD.cChannels - 1 do
        Details[B] := Mute;
        MuteSet := True;
      end;
      mixerSetControlDetails (FMixerHandle, @MCD,
MIXER_GETCONTROLDETAILSF_VALUE);
    end;
  end;
  Inc (A);
end;

end;
end;
end;

```

```

function TAudioMixer.GetMute(ADestination, AConnection: Integer; var Mute:
Boolean):Boolean;

```

```

var

```

```

  MD : TMixerDestination;

```

```

  MC : TMixerConnection;

```

```

  mlcMixerLineControlsMute : TMIXERLINECONTROLS;

```

```

  mcdMixerDataMute : TMIXERCONTROLDETAILS;

```

```

pmcMixerControlMute : PMIXERCONTROL;
pmcdsMixerDataUnsignedMute : PMIXERCONTROLDETAILSBOOLEAN;
mlMixerLine : TMixerLine;
Cntrl:PMixerControl;
Cntrls:TMixerControls;
ML:TMixerLine;
A,B:Integer;
details:array [0..100] of Integer;
ltext:^ListTextArray;
MCDText:TMixerControlDetails;
begin
  Result := False;
  If (not Assigned (FDestinations)) then
    Exit;
  MC := nil;
  Mute := False;
  MD := Destinations[ADestination];
  if MD <> nil then
    begin
      if AConnection = -1 then
        mlMixerLine := MD.Data
      else
        begin
          MC := MD.Connections[AConnection];
          if MC <> nil then
            mlMixerLine := MC.Data
          else
            Exit;
          end;
        end;
      GetMem(pmcMixerControlMute, SizeOf(TMIXERCONTROL));
      GetMem(pmcdsMixerDataUnsignedMute, SizeOf(TMIXERCONTROLDETAILSBOOLEAN));
      with mlcMixerLineControlsMute do
        begin
          cbStruct := SizeOf(TMIXERLINECONTROLS);
          dwLineID := mlMixerLine.dwLineID;
          dwControlType := MIXERCONTROL_CONTROLTYPE_MUTE;
          cControls := 1;
          cbmxcntrl := SizeOf(TMIXERCONTROL);
          pamxcntrl := pmcMixerControlMute;
        end;
      if (mixerGetLineControls(FMixerHandle, @mlcMixerLineControlsMute,
        MIXER_GETLINECONTROLSF_ONEBYTYPE) = MMSYSERR_NOERROR) then
        begin

```

```

with mcdMixerDataMute do
begin
  cbStruct := SizeOf(TMIXERCONTROLDETAILS);
  dwControlID := pmcMixerControlMute^.dwControlID;
  cChannels := 1;
  cMultipleItems := 0;
  cbDetails := SizeOf(TMIXERCONTROLDETAILSBOOLEAN);
  paDetails := pmcdsMixerDataUnsignedMute;
end;

if mixerGetControlDetails(FMixerHandle, @mcdMixerDataMute,
MIXER_GETCONTROLDETAILSF_VALUE) = MMSYSERR_NOERROR then
begin
  Mute := pmcdsMixerDataUnsignedMute^.fValue = 1;
  Result := True;
end;
end
else
begin
  If (AConnection <> -1) then
  begin
    Cntrls := MD.Controls;
    ML := MD.Data;
    If (MC <> nil) AND (Cntrls <> nil) then
    begin
      A := 0;
      while (Result = False) AND (A < Cntrls.Count) do
      begin
        Cntrl := Cntrls[A];
        If (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MIXER) OR
          (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUX) then
          begin
            mcdMixerDataMute.cbStruct := SizeOf(TMixerControlDetails);
            mcdMixerDataMute.dwControlID := Cntrl.dwControlID;
            If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_UNIFORM > 0 then
              mcdMixerDataMute.cChannels := 1
            else
              mcdMixerDataMute.cChannels := ML.cChannels;
            If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_MULTIPLE =
MIXERCONTROL_CONTROLF_MULTIPLE then
              mcdMixerDataMute.cMultipleItems := Cntrl.cMultipleItems
            else
              mcdMixerDataMute.cMultipleItems := 0;
            mcdMixerDataMute.cbDetails := 4;
            mcdMixerDataMute.paDetails := @Details;
          end;
        A := A + 1;
      end;
    end;
  end;
end;
end;

```

```

        mixerGetControlDetails      (FMixerHandle,      @mcdMixerDataMute,
MIXER_GETCONTROLDETAILSF_VALUE);

        GetMem      (ltext,      mcdMixerDataMute.cChannels      *
mcdMixerDataMute.cMultipleItems * sizeof (MIXERCONTROLDETAILS_LISTTEXT));
        MCDText.cbStruct := sizeof (MCDText);
        MCDText.dwControlID := Cntrl.dwControlID;
        MCDText.cChannels := mcdMixerDataMute.cChannels;
        MCDText.cMultipleItems := mcdMixerDataMute.cMultipleItems;
        MCDText.cbDetails := sizeof (MIXERCONTROLDETAILS_LISTTEXT);
        MCDText.paDetails := ltext;
        mixerGetControlDetails      (FMixerHandle,      @MCDText,
MIXER_GETCONTROLDETAILSF_LISTTEXT);

        B := mcdMixerDataMute.cChannels - 1;
        while (B < integer (mcdMixerDataMute.cMultipleItems)) do
        begin
            if (ltext[B].dwParam1 = MC.Data.dwLineID) then
                break;
            Inc (B, mcdMixerDataMute.cChannels);
        end;
        FreeMem (ltext);

        If (B < integer (mcdMixerDataMute.cMultipleItems)) then
        begin
            Result := True;
            Mute := Details[B] <> 0;
            break;
        end;
        end;
        end;
        Inc (A);
        end;
        end;
        end;
        end;

        FreeMem (pmcdsMixerDataUnsignedMute);
        FreeMem (pmcMixerControlMute);
    end;
end;

function TAudioMixer.SetMute (ADestination, AConnection: Integer; Mute:
Boolean): Boolean;
var
    MD : TMixerDestination;
    MC : TMixerConnection;
    mlcMixerLineControlsMute : TMIXERLINECONTROLS;

```

```

mcdMixerDataMute : TMIXERCONTROLDETAILS;
pmcMixerControlMute : PMIXERCONTROL;
pmcdsMixerDataUnsignedMute : PMIXERCONTROLDETAILSBOOLEAN;
mlMixerLine : TMixerLine;
Cntrl:PMixerControl;
Cntrls:TMixerControls;
ML:TMixerLine;
A,B:Integer;
details:array [0..100] of Integer;
ltext:^ListTextArray;
MCDText:TMixerControlDetails;
begin
  Result := False;
  If (not Assigned (FDestinations)) then
    Exit;
  MC := nil;
  MD := Destinations[ADestination];
  if MD <> nil then
    begin
      if AConnection = -1 then
        mlMixerLine := MD.Data
      else
        begin
          MC := MD.Connections[AConnection];
          if MC <> nil then
            mlMixerLine := MC.Data
          else
            Exit;
          end;
        end;

      GetMem(pmcMixerControlMute, SizeOf(TMIXERCONTROL));
      GetMem(pmcdsMixerDataUnsignedMute, SizeOf(TMIXERCONTROLDETAILSBOOLEAN));

      with mlcMixerLineControlsMute do
        begin
          cbStruct := SizeOf(TMIXERLINECONTROLS);
          dwLineID := mlMixerLine.dwLineID;
          dwControlType := MIXERCONTROL_CONTROLTYPE_MUTE;
          cControls := 0;
          cbmxctrl := SizeOf(TMIXERCONTROL);
          pamxctrl := pmcMixerControlMute;
        end;

        if (mixerGetLineControls(FMixerHandle, @mlcMixerLineControlsMute,
MIXER_GETLINECONTROLSF_ONEBYTYPE) = MMSYSERR_NOERROR) then
          begin

```

```

with mcdMixerDataMute do
begin
  cbStruct := SizeOf(TMixerControlDetails);
  dwControlID := pmcMixerControlMute^.dwControlID;
  cChannels := 1;
  cMultipleItems := 0;
  cbDetails := SizeOf(TMIXERCONTROLDETAILSBOOLEAN);
  paDetails := pmcdsMixerDataUnsignedMute;
end;

if Mute then
  pmcdsMixerDataUnsignedMute^.fValue := 1
else
  pmcdsMixerDataUnsignedMute^.fValue := 0;

if
(mixerSetControlDetails(FMixerHandle,@mcdMixerDataMute,MIXER_SETCONTROLDETAILS
F_VALUE) = MMSYSERR_NOERROR) then
  Result := True;
end
else
begin
  If (AConnection <> -1) then
  begin
    Cntrls := MD.Controls;
    ML := MD.Data;
    If (MC <> nil) AND (Cntrls <> nil) then
    begin
      A := 0;
      while (Result = False) AND (A < Cntrls.Count) do
      begin
        Cntrl := Cntrls[A];
        If (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MIXER) OR
          (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUX) then
          begin
            mcdMixerDataMute.cbStruct := SizeOf(TMixerControlDetails);
            mcdMixerDataMute.dwControlID := Cntrl.dwControlID;
            If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_UNIFORM > 0 then
              mcdMixerDataMute.cChannels := 1
            else
              mcdMixerDataMute.cChannels := ML.cChannels;
            If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_MULTIPLE =
MIXERCONTROL_CONTROLF_MULTIPLE then
              mcdMixerDataMute.cMultipleItems := Cntrl.cMultipleItems
            else
              mcdMixerDataMute.cMultipleItems := 0;

```



```

mcdMixerDataMute.cbDetails := 4;
mcdMixerDataMute.paDetails := @Details;
if (mixerGetControlDetails (FMixerHandle, @mcdMixerDataMute,
MIXER_GETCONTROLDETAILSF_VALUE) <> MMSYSERR_NOERROR) then
begin
  Inc (A);
  continue;
end;
if (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUX) then
  For B := 0 to Cntrl.cMultipleItems-1 do
    Details[B] := 0;
If Mute then
begin
  GetMem (ltext, mcdMixerDataMute.cChannels *
mcdMixerDataMute.cMultipleItems * sizeof (MIXERCONTROLDETAILS_LISTTEXT));
  MCDText.cbStruct := sizeof (MCDText);
  MCDText.dwControlID := Cntrl.dwControlID;
  MCDText.cChannels := mcdMixerDataMute.cChannels;
  MCDText.cMultipleItems := mcdMixerDataMute.cMultipleItems;
  MCDText.cbDetails := sizeof (MIXERCONTROLDETAILS_LISTTEXT);
  MCDText.paDetails := ltext;
  mixerGetControlDetails (FMixerHandle, @MCDText,
MIXER_GETCONTROLDETAILSF_LISTTEXT);
  B := mcdMixerDataMute.cChannels - 1;
  while (B < integer (mcdMixerDataMute.cMultipleItems)) do
  begin
    if (ltext[B].dwParam1 = MC.Data.dwLineID) then
      break;
    Inc (B, mcdMixerDataMute.cChannels);
  end;
  FreeMem (ltext);
  If (B < integer (mcdMixerDataMute.cMultipleItems)) then
    Details[B] := 1;
  end;
  if (mixerSetControlDetails (FMixerHandle, @mcdMixerDataMute,
MIXER_GETCONTROLDETAILSF_VALUE) = MMSYSERR_NOERROR) then
  begin
    Result := True;
    break;
  end;
end;
Inc (A);
end;
end;
end;

```

```

end;

FreeMem (pmcdsMixerDataUnsignedMute);
FreeMem (pmcMixerControlMute);
end;
end;

function TAudioMixer.GetPeak (ADestination, AConnection:Integer; var LeftPeak,
RightPeak:Integer): Boolean;
var
  MD : TMixerDestination;
  MC : TMixerConnection;
  mcdMixerDataPeak : TMIXERCONTROLDETAILS;
  pmcMixerControlPeak : PMIXERCONTROL;
{ pmcdsMixerDataSignedPeak : PMIXERCONTROLDETAILSSIGNED;}
  mlMixerLine : TMixerLine;
  A:Integer;
  Cntrls:TMixerControls;
  Details:Array [1..100] of Integer;
begin
  Result := False;
  If (not Assigned (FDestinations)) then
    Exit;
  LeftPeak := 0;
  RightPeak := 0;
  MD := Destinations[ADestination];
  if MD <> nil then
  begin
    if AConnection = -1 then
    begin
      mlMixerLine := MD.Data;
      Cntrls := MD.Controls;
    end
    else
    begin
      MC := MD.Connections[AConnection];
      if MC <> nil then
      begin
        mlMixerLine := MC.Data;
        Cntrls := MC.Controls;
      end
      else
        Exit;
      end;
    end;
  GetMem (pmcMixerControlPeak, SizeOf (TMIXERCONTROL));

```

```

A := 0;
while (A < Cntrls.Count) do
begin
    If (Cntrls[A].dwControlType AND MIXERCONTROL_CT_CLASS_MASK) =
MIXERCONTROL_CT_CLASS_METER then
        break;
    Inc (A);
end;
If A = Cntrls.Count then
begin
    FreeMem(pmcMixerControlPeak);
    Exit;
end;

with mcdMixerDataPeak do
begin
    cbStruct := SizeOf(TMIXERCONTROLDETAILS);
    dwControlID := Cntrls[A].dwControlID;
    cChannels := mlMixerLine.cChannels;
    If (Cntrls[A].fdwControl AND MIXERCONTROL_CONTROLF_MULTIPLE) =
MIXERCONTROL_CONTROLF_MULTIPLE then
        cMultipleItems:=Cntrls[A].cMultipleItems
    else
        cMultipleItems:=0;
    cbDetails := SizeOf(TMIXERCONTROLDETAILSSIGNED);
    paDetails := @Details;
end;
if
(mixerGetControlDetails(FMixerHandle,@mcdMixerDataPeak,MIXER_GETCONTROLDETAILS
F_VALUE) = MMSYSERR_NOERROR) then
begin
    LeftPeak := Details[1];
    if mlMixerLine.cChannels = 2 then
        RightPeak := Details[2]
    else
        RightPeak := LeftPeak;
    Result := True;
end;
FreeMem(pmcMixerControlPeak);
end;
end;
procedure Register;
begin
    RegisterComponents('Samples', [TAudioMixer]);
end;
end.

```