

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

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

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи передачі
мультимедійних даних за технологією IMS”

КБГЗ - 2023

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

Керівник проекту
доктор філософії (PhD)
_____ Дреєва Г.М.
« ____ » _____ 2023 р.
Рецензент _____

Центральноукраїнський національний технічний університет

Центр *Заочної та дистанційної освіти*

Кафедра *Кібербезпеки та програмного забезпечення*

Рівень вищої освіти *магістр*

Галузь знань 12 *“Інформаційні технології”*

Спеціальність 122 *“Комп’ютерні науки”*

Освітньо-професійна (освітньо-наукова) програма *“Комп’ютерні науки”*

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 6 » вересня 2023 року

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

Рябошапці Олександр Олександровичу

(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження та програмна реалізація системи передачі мультимедійних даних за технологією IMS*

2. Керівник роботи *Дреєва Ганна Миколаївна, доктор філософії (PhD)*

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

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

3. Строк подання студентом роботи до захисту *10.12.2023 р.*

4. Мета та завдання випускної кваліфікаційної роботи: *Метою розробки є дослідження та програмна реалізація системи передачі мультимедійних даних за технологією IMS*

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

1. Призначення та область використання.

6. Наукова новизна.

2. Перегляд аналогічних існуючих систем.

7. Економічна ефективність розробленої програми.

3. Опис і обґрунтування проектних рішень.

8. Заходи з охорони праці та техніки безпеки.

4. Етапи програмування системи.

9. Висновки.

5. Впровадження системи в промислову експлуатацію

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

Наукова новизна

1 аркуш

Структурна схема системи

1 аркуш

Функціональна схема системи

1 аркуш

Діаграма процесів

1 аркуш

Блок-схема алгоритму роботи додатку

2 аркуша

Показники економічної ефективності

1 аркуш

6. Консультанти розділів роботи

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

7. Дата видачі завдання « 6 » вересня 2023 р.**КАЛЕНДАРНИЙ ПЛАН**

№ з/п	Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.10.2023 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2023 р.	
3.	Розробка моделі компонента	20.10.2023 р.	
4.	Розробка структур даних	25.10.2023 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2023 р.	
6.	Програмування алгоритмів	10.11.2023 р.	
7.	Розрахунок економічної ефективності	13.11.2023 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2023 р.	
9.	Оформлення ПЗ	17.11.2023 р.	
10.	Попередній захист роботи	10.12.2023 р.	

Дата видачі завдання
« 6 » вересня 2023 р.

Підпис керівника

(прізвище та ініціали)Завдання прийнято до виконання
« 6 » вересня 2023 р.

Підпис здобувача

(прізвище та ініціали)

АНОТАЦІЯ

Рябошапко О.О. Дослідження та програмна реалізація системи передачі мультимедійних даних за технологією IMS. 122 Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2023.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи передачі мультимедійних даних за технологією IMS.

Метою розробки є дослідження та програмна реалізація системи передачі мультимедійних даних за технологією IMS.

Об'єктом дослідження є процес передачі мультимедійних даних за технологією IMS.

Предметом дослідження є методи передачі мультимедійних даних за технологією IMS.

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

Результат роботи – програмна реалізація системи передачі мультимедійних даних за технологією IMS.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ архітектури IBM PC з ОС Windows 10/11.

Програму розроблено в середовищі Delphi 10.4.1.

Ключові слова: комп'ютерні науки, мультимедійні дані, IMS

ABSTRACT

Riaboshapko O.O. Research and software implementation of the multimedia data transmission system using IMS technology. 122 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2023.

In this graduation thesis for the second (master's) level of higher education, software is developed, which is intended for the multimedia data transmission system using the IMS technology.

The purpose of the development is the research and software implementation of the multimedia data transmission system using IMS technology.

The object of the study is the process of multimedia data transfer using IMS technology.

The subject of research is methods of multimedia data transmission using IMS technology.

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

The result of the work is the software implementation of the multimedia data transmission system using IMS technology.

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

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

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

The program was developed in the Delphi 10.4.1 environment.

Keywords: computer science, multimedia data, IMS

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	12
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	12
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	23
2.3 Розгорнута постановка завдання	28
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	30
3.1 Опис функціонування системи	30
3.2 Розробка структурної схеми.....	34
3.3 Розробка функціональної схеми	46
3.4 Розробка діаграми процесів.....	55
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	57
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	57
4.2 Захист розробленого програмного забезпечення.....	69
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	75
6 НАУКОВА НОВИЗНА	77

						ВКРМ-122.23.0080.00.00.ПЗ		
Вим	Арк.	№ докум.	Підп.	Дата				
Розроб.	Рябошапко О.О.				Дослідження та програмна реалізація системи передачі мультимедійних даних за технологією IMS	Літ.	Аркуш	Аркушів
Перев.	Дресва Г.М.					М	1	117
Н.контр.	Коваленко А.С.					ЦНТУ КН-22МЗ		
Затв.	Смірнов О.А.							

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	78
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	78
7.2 Розрахунок трудомісткості розробки програмної продукції.....	80
7.3 Визначення чисельності виконавців і планового фонду зарплати.....	82
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника.....	86
7.5 Визначення собівартості розробки та ціни програмної продукції.....	90
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	94
7.7 Визначення експлуатаційних витрат.....	94
7.8 Визначення економічної ефективності програмної продукції.....	96
7.9 Висновок.....	98
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	99
8.1 Вступ.....	99
8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером.....	101
8.3 Розробка заходів з умов поліпшення охорони праці.....	102
8.4 Пожежна безпека.....	103
8.5 Розрахунок занулення.....	107
9 ОСНОВНІ ВИСНОВКИ.....	110
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	112

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
IMS	– Мультимедійна підсистема на основі протоколу IP.
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
IP	– 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

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

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

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

ВСТУП

Актуальність теми. IMS (IP Multimedia Subsystem) – мультимедійна підсистема на основі протоколу IP. Спочатку архітектура розроблялась тільки як мультимедійна платформа надання послуг (SDP – Service Delivery Platform). Але пізніше перетворилася на архітектуру, повністю контролює з'єднання і працює з різними мережами доступу.

Концепція IMS визначає засновану на загальнопоширених протоколах сімейства TCP/IP архітектуру надання сервісів (послуг), що забезпечує керування сеансами зв'язку й доставку в рамках цих сеансів будь-яких типів інформації – мови, даних, відео, мультимедіа. Принципово важливо те, що в системах, що відповідає концепції IMS, послуги можуть надаватися різними сервіс-провайдерами й доставлятися до користувачів по різних (провідних і бездротовим) мережах доступу.

У мережі IMS користувач може підписатися на пакет послуг, зареєструвавши для їхнього одержання кілька терміналів з різними характеристиками, адресами й типами підключень. Це можуть бути: домашній ПК, підключений до Інтернету через DSL-лінію або будинкову мережу Ethernet; мобільний телефон із включеним сервісом GPRS; ноутбук або кишеньковий ПК, “який виходить на зв'язок” через хот-споти Wi-Fi. Кожний із цих терміналів реєструється окремо, але всі вони асоціюються з одним користувачем, що задає правила, по яких вхідні комунікаційні виклики будуть розподілятися між різними терміналами.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи передачі мультимедійних даних за технологією IMS.

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

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

- Огляд існуючих систем передачі мультимедійних даних за технологією IMS.
- Дослідження системи передачі мультимедійних даних за технологією IMS.
- Програмна реалізація системи передачі мультимедійних даних за технологією IMS.

Об'єктом дослідження є процес передачі мультимедійних даних за технологією IMS.

Предметом дослідження є методи передачі мультимедійних даних за технологією IMS.

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

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

- Удосконалено метод передачі мультимедійних даних за технологією IMS.
- Розроблено вітчизняний продукт передачі мультимедійних даних за технологією IMS, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі передачі мультимедійних даних за технологією IMS.

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

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

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

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи передачі мультимедійних даних за технологією IMS, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

КБПЗ – 2023

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Мультимедійність архітектури дає можливість оператору надавати різноманітні послуги абонентам, підвищуючи тим самим середню виручку (ARPU). А використання в основі протоколу IP дає оператору можливість побудувати гнучку мережу з низькими операційними витратами. Крім того, в основі лежить горизонтальна архітектура, на відміну від традиційної – вертикальної.

Говорячи про “загальнопоширені протоколи із сімейства TCP/IP”, на яких базується IMS, у першу чергу необхідно виділити SIP. Чому саме SIP виявився практично ідеальним інструментом рішення тих завдань, які поставили перед собою розроблювачі ISM? Цей відносно простий протокол призначений саме для керування сеансами зв'язку (ініціація, модифікація, завершення), причому він дозволяє будь-якому числу користувачів динамічно підключатися до сеансу й виходити з нього – звідси широкі можливості по організації всякого роду конференцій. Не менш важливо й те, що SIP дає можливість динамічно – знов-таки в рамках існуючого сеансу зв'язку! – підключати нові типи інформації; наприклад, сеанс зв'язку можна почати з текстового чата, потім додати голосовий зв'язок, а потім при необхідності й відеокартинку. Нарешті, засобу SIP здатні при ініціації або модифікації сеансу зв'язку враховувати характеристики каналу доступу й терміналу кожного користувача й задіяти їхнім оптимальним образом. Скажемо, для абонента відеотерміналу, підключеного по широкополосному каналі, будуть доступні всі види зв'язку, аж до відео високого дозволу, а для користувача старенького мобільного телефону – тільки базові (голосовий зв'язок і SMS).

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

1.2 Область застосування

Основною організацією, у документах якої прописана архітектура IMS, є організація 3GPP (3rd Generation Partnership Project). Споконвічно метою 3GPP було створення специфікацій на систему мобільного зв'язку третього покоління (3G), яка б стала подальшим розвитком мереж GSM. Специфікації 3GPP публікуються в так званих релізах, перший з яких з'явився в 1999 р. і відомий як Release 99. Що цікаво, у ньому передбачалася побудова мереж 3G на основі технологій TDM і ATM і взагалі не згадувалися ніякі IP-елементи. У специфікаціях Release 4 (2001 р.) логічні функції по керуванню викликами відділені від транспорту й запропонована архітектура, у якій передбачене розподіл на логічні рівні, але знов-таки немає поняття IMS. Воно з'явилося в документах Release 5 на початку 2003 р. Там визначена концепція IMS, застережене використання протоколів SIP і Diameter, функцій CSCF і HSS, базових принципів QoS. Release 5 орієнтований на мережі 3G UMTS.

Остання із завершених серій документів – Release 6 – поширює концепцію IMS на інші типи мереж, включаючи мережі WLAN. Помічу, що ряд піонерських проектів, що просуваються як “IMS-проекти”, реалізований не в мережах 3G, а в мережах фіксованого зв'язку, де теж застосовні такі базові поняття IMS, як CSCF і HSS. Зараз організація 3GPP працює над специфікаціями Release 7, у них будуть, зокрема, застережені важливі моменти, пов'язані з контролем правил надання послуг (policy control) і наскрізним забезпеченням якості обслуговування (QoS). Хоча остаточна версія Release 7 з'явиться не раніше кінця 2006 – початку 2007 р., не виключено, що ряд нових елементів будуть реалізовані в конкретному встаткуванні раніше цього строку.

Другою по значимості “законотворчою” організацією в сфері IMS є європейський інститут ETSI. Для нас, крім глобальної значимості ініціатив ETSI, важливо ще й те, що рекомендації саме цього інституту звичайно адаптують для застосування в Україні. У вересні 2003 р. у ньому була організована робоча група

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

Tispan, метою якої став виробіток специфікацій для мультисервісних IP-мереж з підтримкою мультимедійних функцій, мобільності абонентів і сумісністю із традиційними мережами зв'язку. Експерти Tispan надійшли мудро: вони не стали винаходити велосипед, а взяли за основу IMS-архітектуру, певну в документах 3GPP Release 6. Наступне таке ж розумне рішення Tispan полягало в тому, що всі додаткові розширення, необхідні для сумісності нових систем із традиційними провідними інфраструктурами, ця робоча група вирішила надавати на узгодження в організацію 3GPP. Таким чином, остання залишилася остаточним арбітром у справі рішення всіх питань, пов'язаних з IMS.

Група Tispan розробила два основних механізми, що перекидають “мости” між IMS і традиційної ТфОП. Перший називається “емуляція ТфОП” (PSTN emulation): нові системи “наслідують” класичним телефонним системам, обслуговуючи звичайні аналогові телефонні апарати. Другий – “імітація ТфОП” (PSTN simulation): відбувається моделювання ТфОП-подібних послуг при обслуговуванні IP-телефонів, програмних телефонів і т.п. Зрозуміло, що імітацію реалізувати значно простіше, ніж емуляцію. Важливим внеском Tispan у розвиток технологій IMS стала також розробка підсистеми керування ресурсами й доступом (Resource and Admission Subsystem). Ця підсистема визначає можливість установа сесансу зв'язку (наприклад, з'ясовує, немає чи перевантаження в пікові годинники) і контролює використання транспортних ресурсів на границі між мережами доступу й магістраллю. Специфікації Tispan визначають можливість використання IMS як у бездротові (3G UMTS, WLAN), так і в провідних мережах (доступ через ЛОМ, xDSL).

Серед великої кількості інших організацій по стандартизації, що проявляють активність “навколо IMS”, відзначу альянс OMA (Open Mobile Alliance), що прагне забезпечити сумісність нових послуг, що розгортаються в мережах мобільного зв'язку. Найвідомішим на сьогодні проектом OMA є розробка специфікації Push-to-talk Over Cellular (POC), про її підтримку заявили багато виробників устаткування. Робочі групи OMA трудяться й над багатьма

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

іншими, дуже важливими для майбутнього IMS прикладними службами: миттєвого обміну повідомленнями й контролю статусу абонента, обліку відомостей про його місцезнаходження (location) і ін. ОМА приділяє підвищену увагу сумісності нових сервісів при взаємодії їх через інтерфейси й інші елементи IMS.

З короткого огляду проектів по стандартизації IMS треба важливий вивід: специфікації на IMS остаточно ще не сформовані – отже, говорити про стовідсотково стандартні рішення IMS ще рано. Щоб претендувати на використання “лейбла” IMS, мережа повинна щонайменше містити такі елементи, як CSCF і HSS, і забезпечувати відкритий інтерфейс до прикладних систем.

Таким чином, виходячи з вищеперахованого, дослідження та програмна реалізація системи передачі мультимедійних даних за технологією IMS, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

КБПЗ-2023

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

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

Відповідно до попередніх оцінок дослідницької компанії Venture Development Corporation (VDC), обсяг ринку основних інфраструктурних систем IMS в 2023 р. склав порядку 560 млн дол. Приблизно половина цієї суми доводиться на сервери додатків, четверта частина – на системи керування сеансами зв'язку (Call Session Control Function – CSCF), або, попросту говорячи, SIP-сервери. У фінансовому аналізі VDC виділені ще два типи елементів – системи обробки медіапотоків (Media Resource Function – MRF) і абонентські бази даних (Home Subscriber Server – HSS), – які забезпечили відповідно близько 17 і 4% заявленої вище суми продажів.

На найближчі три роки аналітики VDC прогнозують стабільний ріст ринку встаткування IMS у середньому на 67% щорічно. При цьому швидше всього (на 185%) будуть рости продажі серверів HSS, тобто тих елементів, які сьогодні становлять найменшу частку в продажах систем IMS.

Часто до систем IMS відносять і різного роду медіа– і сигнальні шлюзи, керування якими здійснюється серверами CSCF з використанням протоколів H.248 і/або SIP. Насправді впровадження таких шлюзів прямо не пов'язане з реалізацією проектів IMS – вони використовуються із самого початку впровадження технологій пакетної телефонії скрізь, де потрібно забезпечити сполучення із існуючими мережами зв'язку. За своїм обсягом ринок таких шлюзів значно перевершує ринок інфраструктурних компонентів IMS. За різними оцінками, в 2023 р. він склав порядку 1 млрд дол.

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

Важливий елемент, без якого немисливо масове впровадження IMS-систем і послуг, – це термінали, орієнтовані на роботу в IMS-мережах. Такі термінали, скажемо мобільні трубки, повинні щонайменше “уміти” ініціювати й обробляти SIP-запити. На жаль, поки ринок IMS-терміналів фактично відсутній, і обсяги продажів у цьому сегменті на порядок нижче обсягів продажів відповідного інфраструктурного встаткування.

Навесні 2023 р. компанія ABI Research провела масштабне дослідження, присвячене IMS. У її опитуванні взяли участь фахівці як постачальників послуг зв'язку, так і виробників устаткування й програмного забезпечення. На думку респондентів ABI, трійка компаній, що мають найбільш сильні позиції в області IMS-рішень, виглядає так: Ericsson, Nokia, Lucent. Також у число провідних постачальників IMS-рішень увійшли компанії Siemens, Motorola, Alcatel, Nortel, Cisco, NEC.

Alcatel

За твердженням фахівців Alcatel, пропоноване компанією комплексне рішення IMS повністю відповідає специфікаціям Release 5/6/7 3GPP (3rd Generation Partnership Project) – основної організації, що відає питаннями стандартизації IMS.

Системи керування сеансами зв'язку (CSCF) цієї компанії стали подальшим розвитком відомого комутатора Alcatel 5020 Softswitch. Для мереж фіксованого й мобільного зв'язку компанія пропонує контролер 5020 CSC (Call and Session Controller). Для спрощення перекладу магістралей мобільних мереж із традиційних технологій тимчасового ущільнення (TDM) на пакетні технології призначений сервер 5020 WCS (Wireless Call Server); це рішення з'явилося в портфелі Alcatel після придбання нею в березні 2004 р. компанії Spatial Wireless.

У якості ще одного ключового елемента архітектури IMS – абонентської бази даних – в Alcatel є система 1430 HSS.

Рішення Alcatel прикладного рівня побудовані на базі архітектури Open Services Delivery Environment. Платформа 8690 OSP (Open Services Platform)

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

забезпечує все необхідне для розробки й розгортання нових послуг, а також функції керування послугами й контентом. Сервер додатків Alcatel 5350 Application Server надає стандартні інтерфейси для “підключення” зовнішніх сервісів.

Повноцінні технічні тестові випробування IMS-систем Alcatel проведені в декількох європейських операторів. Компанія Alcatel поставляє комплексні рішення IMS, включаючи всі необхідні компоненти.

Cisco

У березні 2023 р., інженер-консультант Cisco Олександр Фелижанко визначив рішення компанії на основі платформи Cisco Call Session Control Platform як IMS-Lite. Воно побудовано відповідно до принципів IMS, у ньому присутні основні компоненти й реалізовані найбільш критичні інтерфейси IMS для найшвидшого розгортання операторами нових послуг (наприклад, інтерфейс ISC для зв'язку із серверами додатків).

Названа вище платформа Cisco за своєю функціональністю відповідає логічним елементам IMS CSCF. Вона складається з наступних компонентів: Cisco Service Engine, Cisco Edge Proxy і Cisco Name Resolution Server – всі вони побудовані на основі протоколу SIP. У завдання сервісної системи Service Engine входить інтеграція послуг, визначення місця розташування (location) абонентів і керування відомостями про їх. Компонент Edge Proxy виконує адаптацію сигналізації SIP при підключенні різних типів мереж доступу. Нарешті, сервер Name Resolution служить для маршрутизації викликів між операторами SIP, а також при взаємодії із ТфОП і з мережами рухливого зв'язку.

Для керування медіашлюзами є добре відомий комутатор Cisco PGW 2200 Softswitch, а обробка медіапотоків може здійснюватися на базі голосових шлюзів Cisco AS5000. Як абонентська база даних (HSS) фірма Cisco застосовує систему компанії-партнера.

Один із практичних прикладів впровадження IMS-рішення Cisco – це мережа Sprint PCS. За допомогою рішення Cisco і сервера додатків компанії-

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

партнера в мережі CDMA реалізована послуга “натисни й говори” (Push-to-Talk) на базі SIP. Щодня обслуговуються понад 750 тис. абонентів і обробляються понад 9 млн запити SIP INVITE. В експлуатації цей сервіс перебуває з осені 2003 р. Інший приклад – інсталяція на мережі великого оператора Comcast, що пропонує різні мультимедійні послуги, включаючи відеотелефонію. У декількох операторів рішення Cisco працює в режимі досвідченої експлуатації.

Говорячи про перспективи розвитку IMS, Олександр Фелижанко відзначив, що “перше, чого варто очікувати, – це завершення процесу стандартизації системи IMS (всі сьогоднішні рішення – передстандартні), друге – це поява нових додатків і послуг на її основі”.

Ericsson

Компанія Ericsson є одним з піонерів в області IMS, що відповідають початку розробки вона ще в 2002 р. Пропоноване сьогодні рішення Ericsson IMS Common System 3.0 складається з декількох блоків. У базовий блок (IMS Core) входять елементи, відповідальні за керування сеансами (CSCF), зберігання й обробку абонентських даних (HSS), обробку медіапотоків (MRF), а також за перетворення адрес і доменних імен (IPWorks). Блок елементів IMS Enablers забезпечує взаємодія із серверами додатків, блок Interworking – з різними типами мережного транспорту, а блок Support System містить необхідні системи експлуатації (OSS), автоматичного конфігурування, тарифікації й т.п.

Апаратно-програмною платформою систем Ericsson IMS служить технічне рішення Telecom Server Platform (TSP), що використовує стандартні модулі 3-PCI. Програмне забезпечення TSP працює поверх ОС Linux (для статичних процесів уведення-виводу й керування вузлами) і фірмової ОС DICOS (вона оптимізована для динамічних процесів обробки трафіку й проведення транзакцій баз даних у реальному часі). Функція географічного резервування вузлів забезпечує безперервність надання послуг навіть у випадку серйозних аварій в енергосистемах, при пожежах, землетрусах і інших катаклізмах.

За станом на квітень 2003 р. компанія Ericsson уже підписала 28 контрактів на комерційну й досвідчену експлуатацію рішень IMS. Вони

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

встановлені на мережах GSM/GPRS, WCDMA, CDMA2000 і мережах фіксованого зв'язку й забезпечують реалізацію послуг Push-to-Talk, VoIP, WeShare, IP Centrex і ін. Серед клієнтів Ericsson в області IMS такі оператори, як Telia Sonera, Telefonica, TDC, Telecom Italia Mobile.

Відповідаючи на питання, експерти Ericsson відзначили, що, на їхню думку, IMS представляється сьогодні найкращим підходом до конвергенції мобільного й провідний зв'язка. Та сама IMS-платформа з її універсальною архітектурою може бути використана для надання послуг у мережах 2.5G, 3G, і у фіксованих мережах. Компанія Ericsson планує надалі вдосконалювати своє IMS-рішення, збільшуючи в першу чергу спектр абонентських додатків.

Huawei

Як відзначив заступник директора офісу Huawei по системах комутації й доступу, у цей час “стандартизація IMS ще далеко не закінчена, немає ні широкого застосування IPv6 (без якого неможливо повне використання переваг IMS), ні широкого вибору SIP-терміналів (SIP, як відомо, – основний протокол в IMS), тому поки про IMS можна говорити тільки в майбутньому часі, як про перспективу розвитку”.

Проте в Huawei є рішення IMS (з огляду на вищесказане, його, як і рішення інших компаній, очевидно, точніше називати яке відповідає задуму IMS), причому воно відповідає рекомендаціям ETSI TISPAN і активно розвивається. Відповідно до присланих нас інформації, для керування сеансами зв'язку (елементи CSCF) компанія пропонує свою розробку CSCF3300, а як база даних з абонентською інформацією (HSS) – систему HSS9820. Управляти медіашлюзами “здатний” програмний комутатор Soft3000. У портфелі розробок компанії Huawei є й реалізації всіх інших логічних елементів IMS.

Nokia

PR-служба компанії Nokia провела велике дослідження, що присвячено оцінці користувачами перспектив впровадження основних IMS-сервісів. До них експерти віднесли сервіси обміну повідомленнями в режимі реального часу (Instant Messaging – IM), спільного використання відео (video sharing) і іншого

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

контенту (content sharing), послуги з контролю статусу абонента (presence) і послуги типу Push-to-Talk. Аналіз показав, що від 33 до 43% опитаних користувачів мобільних телефонів (опитування проводилися в Німеччині, Великобританії, Сінгапурі й США) протягом найближчих двох років готові почати активне використання перерахованих нових послуг. По потенційних доходах серед цих послуг у Німеччині й США на першому місці виявився сервіс Push-to-Talk, у Великобританії – спільне використання відео, а в Сінгапурі – контроль статусу абонента.

На жаль, представники Nokia нічого нам не повідомили про суть свого рішення IMS. На Web-сайті компанії у відкритому доступі виявилися лише самі загальні матеріали по IMS. Проте нам удалось з'ясувати, що в компанії є реалізації всіх основних компонентів IMS. Зокрема, функціональність CSCF і HSS реалізована на апаратній платформі FlexiServer. Як сервери додатків Nokia пропонує як власні системи, так і сервери свого партнера – компанії Ubiquity. Не дивно, що Nokia проявляє більшу активність у частині розробки програмних клієнтів для мобільних трубок. Зокрема, компанія пропонує інструментальні засоби розроблювача (SIP SDK) і прикладний програмний інтерфейс (SIP API) з метою створення додатків для своїх смартфонів.

Lucent

Для керування сеансами зв'язку (функціональність CSCF) компанія Lucent пропонує свій “менеджер” Lucent Session Manager. Він же здатний забезпечити обробку медіапотоків (MRF) і керування правилами передачі інформаційних потоків (PDF). Для зберігання абонентських даних і керування ними призначений сервер Lucent Unified Subscriber Data Server (USDS). Він може зберігати в одній базі даних інформацію про місце положення абонента, його особисті дані, настроювання безпеки й детальну інформацію про профіль. Компанія Lucent пропонує й інші елементи, передбачені концепцією IMS, включаючи працюючі по периметрі мережі системи керування медіашлюзами й самі шлюзи.

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

На рівні додатків в Lucent є набір технологій, сервісів, порталів і інших елементів, які дозволяють надати абонентам “єдину й просту систему доступу до всіх додатків і сервісів”. Серед цих елементів – системи Lucent Communications Manager, Lucent Feature Server 5000/3000/ 2500, серія платформ і додатків MiLife. Для керування наданням послуг у реальному часі служить програмний пакет Lucent VitalSuite.

Що відрізняє пропозицію Lucent від рішень інших компаній, так це в першу чергу набір розроблених Bell Labs технологій Lucent IMS Service Enhancement Layer. Так, наприклад, технологія DataGrid дозволяє завжди відслідковувати місце розташування абонента, технологічні можливості поточного каналу, яким він користується, і, виходячи із заданих політик, організовувати відповідний доступ до запитуваних сервісів. Технологія GUPster дає абонентові можливість працювати в різних мережах, зберігаючи єдиний інтерфейс і користувальницький профіль. Ще одна технологія, що серйозно відрізняє рішення Lucent IMS від продуктів конкурентів, – це Vortex. Завдяки їй користувач може сам визначити переваги й правила для послуг зв'язку, які негайно будуть застосовані в системі IMS.

Motorola

У документах, представлених нам компанією Motorola, зазначено, що її рішення IMS реалізоване у відповідності зі стандартами 3GPP і OMA (Open Mobile Alliance). Компанія брала участь у більшості тестів, проведених альянсом OMA при розробці й налагодженні специфікації Push-to-talk Over Cellular (PoC). За даними на квітень 2023 р., в активі Motorola – 44 контракти в 33 країнах на реалізацію комерційних проектів PoC на основі системи Motorola IMS.

Саме рішення Motorola IMS базується на платформі softswitch і відповідно до концепції IMS не залежить від специфіки мереж доступу. У бездротових мережах підтримуються як традиційні інтерфейси EDGE, UMTS, CDMA, так і високошвидкісні технології, що активно розвиваються в цей час, HSDPA, EV-DO, WiMAX; у провідних інфраструктурах підтримується, зокрема, технологія

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

DOCSIS. Модульна серверна платформа Motorola IMS Control Server реалізує функціональність базових елементів IMS по керуванню сеансами зв'язку (CSCF) і абонентської бази даних (HSS), а також функціональність сервера обробки медіапотоків і медіашлюза.

Пропоновані компанією Motorola сервери додатків підтримують не тільки SIP, але й такі традиційні прикладні інтерфейси, як Camel і Parlay. Спеціальний сервер SCIM (Service Capability Interaction Manager) забезпечує керування взаємодією між серверами додатків і різними “драйверами” послуг (service enabler). Крім PoC, компанія Motorola у своєму пакеті рішень IMS пропонує різні послуги Push-to-X (push to view, push to video, push to text...), потокове відео й відеотелефонію, мультимедійні пошту й зв'язок-сполучення-конференц-зв'язок, ігрові сервіси та ін.

NEC

Портфель рішень IMS компанії NEC складається з п'яти базових блоків: керування SIP-сеансами зв'язку; додатка; засобу взаємодії з різними мережами; шлюзи до існуючої інфраструктури; системи узгодження й мережного керування.

У перший блок входить властиво сервер керування викликами й сеансами зв'язку MX 5840-CS, що підтримує обидві версії протоколу IP (IPv4 і IPv6) і, відповідно до термінології IMS, реалізує функції CSCF. Також у цьому блоці – система MX 5640-HS, що здійснює керування абонентськими даними (функціональність HSS). У блок “дodatка” входять кілька серверів додатків, які реалізують сервіси контролю присутності (MX 7840-PR), “натисни й говори” (MX 7840-PT), IM-пошти (MX 7840-CT), а також інші сервіси, у тому числі ігрові й пов'язані з обміном відеоінформацією.

Для взаємодії з іншими мережами є медіашлюзи MX 5240-MG і контролер медіашлюзов MX 5840-MS. У цьому ж блоці – SIP-Шлюз MX5840-SI, що забезпечує сполучення з іншими мережами SIP, виконуючи функції прикордонного контролера, міжмережного екрана, системи трансляції адрес (NAT) та ін. Технічні рішення, називані фахівцями NEC “шлюзами до існуючої

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

інфраструктури”, дозволяють поширити IMS-сервіси (наприклад, по миттєвому обміні повідомленнями) на термінали, не підтримуючий протокол SIP. За допомогою цих рішень оператори зв'язку зможуть розширити базу потенційних клієнтів послуг IMS, що особливо важливо на початковій стадії впровадження нової технології.

Будучи одним із провідних постачальників терміналів, компанія NEC розробила програмний клієнт IMS, що був представлений на початку 2023 р.

Nortel

Як і інші компанії, що мають традиційно сильні позиції на ринку провідних і бездротових систем зв'язку, Nortel особливо підкреслює перспективи конвергенції. Не випадково своє IMS-рішення компанія називає “конвергентними мультимедійними сервісами” – Converged Multimedia Services (CMS).

Основні елементи системи CMS мають ті ж позначення, що й відповідні логічні блоки в специфікаціях IMS: це контролер керування сеансами Nortel Call Session Controller (CSCF) і абонентська база даних Nortel Home Subscriber Server (HSS). Представники Nortel підкреслюють, що контролер CSCF став результатом масштабних п'ятирічних досліджень і досвідчених розробок, проведених у рамках програми по розвитку сервера мультимедійних комунікацій (Multimedia Communication Server – MCS) – добре відомого продукту компанії.

Для взаємодії із ТфОП і іншими традиційними мережами зв'язку компанія Nortel пропонує медіашлюзи й відповідний контролер. Особливо відзначимо систему Nortel Policy Controller, що покликана забезпечити наскрізну гарантію якості обслуговування (QoS), що є найважливішим завданням для конвергентних мереж операторського класу. Як сервер додатків, як один з можливих варіантів, Nortel пропонує використовувати свій продукт MCS5200.

Для розвитку рішення CMS компанія Nortel сформувала велику “екосистему” партнерів, куди, зокрема, входять IBM, Sonim Technologies, Ubiquity, Metasolv. Наприклад, разом з Sonim вона вивела на ринок рішення по наданню послуг PoC, причому це рішення відповідає стандарту, прийнятому

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

альянсом OMA. Технічні рішення Metasolv забезпечують систему CMS засобами збору білінгової інформації в реальному масштабі часу.

Siemens

Архітектура рішення IMS@vantage має модульну структуру й повністю відповідає рекомендаціям Release 5 організації 3GPP. На думку заступника директора центра компетенції по технічній підтримці збуту підрозділу “Мобільні мережі зв'язку” Siemens Communications Володимира Шапорова, поетапне впровадження компонентів IMS можна починати вже зараз, наприклад у мережі з підтримкою GPRS і протоколом IPv4. Для цього досить впровадити тільки елементи CSCF (контролер Siemens CFX-5000) і HSS (сервер Siemens CMS-8200) як обов'язкові компоненти для базових SIP-послуг – PoC, IM, чати, контроль присутності. Надалі оператор може розширити систему до повноцінного рішення IMS з підтримкою протоколу IPv6 і маси нових послуг (додаткові сервіси мовного зв'язку, відео/мультимедіаконференції та ін.), характерних для мереж 3G/ W-CDMA і стандарти 3GPP Release 5.

У рішенні IMS@vantage передбачено відкрите середовище розробки додатків на базі стандартних інтерфейсів ISC і Sh системи IMS. Компанією Siemens запущений проєкт IDP (IMS Developer Project), у рамках якого розроблювачам пропонується інструментарій SDK, приклади додатків, емулятор роботи мережі IMS (для тестування додатків) і т.д. Відповідаючи на наше питання про сполучення із системами інших фірм, Володимир Шапоров сказав, що в цей час (квітень 2023 р.) проводяться тести на сумісність із системою Ericsson IMS, планується тестування з рішеннями всіх провідних виробників.

За станом на квітень 2023 р. компанія Siemens підписала п'ять комерційних контрактів на розгортання системи IMS, у тому числі проєкт FMC (Fixed Mobile Convergence) для оператора KPN у трьох країнах, і провела більше 30 тестових інсталяцій. Рішення Siemens IMS уже протестовано й тестується в Україні різними операторами. Вони досліджують роботу IMS-систем по наданню як згаданих вище сервісів – PoC, IM, чати, контроль присутності, – так і їхніх

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

комбінацій – послуги “ фоторазго-злодій” (PhotoTalk), мультичат (Multichat), обмін анімованими повідомленнями (Animated Instant Voice Messaging) і ін.

ZTE

Відповідно до присланих нас із компанії ZTE інформації, вона готова запропонувати реалізації всіх основних елементів IMS, включаючи систему керування сеансами (CSCF), базу абонентських даних (HSS), медіашлюзи, їхні контролери й сервери додатків.

Зауваження фахівців ZTE із приводу перспектив впровадження систем IMS, на наш погляд, показують їхнє чітке розуміння ситуації на вітчизняному телекомунікаційному ринку. Вони відзначають, що українські оператори тільки починають розвертати мережі NGN класу 4 для транзитної комутації або розглядають варіанти установки систем softswitch у ядрі мобільної мережі, але поки вони далекі від конвергенції всіх служб в одну мережу з єдиним набором послуг. Ще один фактор, що відсуває впровадження IMS, – це недостатня цифровізація мереж (як фіксованих, так і мобільних). Крім того, у мобільному зв'язку поки недостатньо високі швидкості передачі даних. Компанія ZTE пропонує операторам поетапний перехід до IMS, починаючи із впровадження встаткування softswitch із закладеними багатими можливостями, а потім поступове впровадження нових послуг з одночасним додаванням у мережу необхідних елементів. Аналіз рішень показує, що системи IMS є на сьогодні однією із самих стрімко, що розвиваються, областей телекомунікацій. У своїх оглядах компанії приділили увагу не тільки перевагам архітектури IMS (відкритість, незалежність від специфіки технологій доступу та ін.) і перевагам своїх рішень, але й конкретним послугам, реалізованим на їхній основі. Також порадював і реалістичний підхід до оцінки перспектив впровадження систем IMS. Все це змушує повірити, що IMS не залишиться лише “паперовою концепцією”, а стане реальним інструментом, що дозволить операторам і сервісам-провайдерам при оптимізації витрат запропонувати нові, дійсно цікаві користувачам послуги й одержати додаткові джерела прибутку.

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Embarcadero Delphi, раніше Borland Delphi і Codegear Delphi, – інтегроване середовище розробки ПЗ для Microsoft Windows, Mac OS, iOS і Android мовою Delphi (що раніше носила назву Object Pascal), створена спочатку фірмою Borland і на даний момент приналежна й розроблювальна Embarcadero Technologies. Embarcadero Delphi є частиною пакета Embarcadero RAD Studio і поставляється в чотирьох редакціях: Community (поширюється безкоштовно й має обмежену ліцензію на використання в комерційних цілях), Professional, Enterprise і Architect.

Delphi 10.4 Sydney

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

Основні можливості Delphi 10.4.1:

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

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

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

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

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

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

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

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

– Підтримка Metal Driver GPU для macOS і iOS.

– Вбудований Fmxlinux.

– Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.

Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

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

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

– Додані оновлені драйвери для FireBird, PostgreSQL і SQLite.

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

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

RAD Studio 10.4 Короткий огляд:

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

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

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

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

Істотне поліпшення Delphi Code Insight

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

Delphi Custom Managed Records

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

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

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

Єдине керування пам'яттю

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

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

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

Розширена підтримка бібліотек C++

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

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

Win 64-відладник і збирач для C++

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

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

Підвищення якості й швидкодії інструментів

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

Змінені стилі VCL для High DPI

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

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

Нові High DPI стилі й стилізація окремих VCL компонент

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

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

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

Поліпшена кроссплатформеність

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

Оновлений менеджер пакетів Getit

Менеджер пакетів Getit в IDE був значно вдосконалений.

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

Універсальний інсталятор для установки Online і Offline

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

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

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випуск кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

забезпечення, яке призначено для системи передачі мультимедійних даних за технологією IMS.

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

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

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

Розробити функціональну та структурну схеми системи;

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

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

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

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

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

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

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Після того як ідея передачі голосу за допомогою мереж з пакетною комутацією і протоколу IP увінчалася успіхом, оператори всерйоз задумалися про зміну існуючих телекомунікаційних мереж. Розробкою взаємоз'єднання телекомунікаційних мереж з технологіями пакетної комутацією зайнялася група 3GPP. Появою пакетного доступу в стільникових мережах прийнято вважати 99 реліз 3GPP (3GPP R99) в якому з'явилася підтримка PS домену в існуючих мобільних мережах на базі технології GPRS. Але роботи продовжувалися і вже в 4 релізі, комутація абонентів лежала на програмному комутаторі SoftSwitch (R4 3GPP). П'ятий реліз (R5 3GPP), ознаменував появу архітектури IMS, правда спочатку тільки в ролі підсистеми надання мультимедійних послуг, але надалі (R5 3GPP) архітектура стала ключовою і комутацію абонентів перемкнула на себе.

3GPP

Спочатку ідеєю побудови мобільної мережі на базі IP технологій займалася група 3G.IP. Саме вона розробила технологію GPRS, яка згодом поклала початок розробці архітектури IP мережі. Згодом була зібрана робоча група 3GPP, яка і представила в 2001 році реліз 4 (спочатку названий Release 2000) в якому з'явилися елементи ALL-IP архітектури. П'ятий реліз представив початковий варіант архітектури названої IMS і додалася технологія високошвидкісної пакетної передачі даних (HSDPA). У шостому релізі в архітектуру IMS були введені зміни, а також з'явилася підтримка Wireless LAN мереж. Завдяки роботі групи TISPAN, сьомий реліз 3GPP додав підтримку фіксованих мереж.

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

Також у даний момент робочою групою 3GPP2 розробляється підтримка технології CDMA2000 в архітектурі IMS.

В якості основного протоколу був обраний протокол устанавлення з'єднань (SIP). Важлива особливість SIP – розширюваність, яка полягає в можливості доповнення протоколу новими функціями за рахунок додавання нових заголовків та повідомлень, що дозволяє додавати новий функціонал в мережу без зміни протоколу.

Базовими елементами опорної мережі архітектури IMS є:

– CSCF (Call Session Control Function) – елемент з функціями управління сеансами і маршрутизацією, складається з трьох функціональних блоків:

– P-CSCF – посередник для взаємодії з абонентськими терміналами. Основні завдання – аутентифікація абонента та формування облікового запису;

– I-CSCF – посередник для взаємодії із зовнішніми мережами. Основні завдання – визначення привілеїв зовнішнього абонента з доступу до послуг, вибір відповідного сервера додатків і забезпечення доступу до нього;

– S-CSCF – центральний вузол мережі IMS, обробляє всі SIP-повідомлення, якими обмінюються кінцеві пристрої.

– HSS – сервер домашніх абонентів, є базою для користувача даних і забезпечує доступ до індивідуальних даних користувача, пов'язаними з послугами. У разі якщо в мережі IMS використовується кілька серверів HSS, необхідно додавання SLF (Subscriber Locator Function) який займається пошуком HSS з даними конкретного користувача.

– BGCF – елемент керуючий пересиланням викликів між доменом комутації каналів і мережею IMS. Здійснює маршрутизацію на основі телефонних номерів і вибирає шлюз в домені комутації каналів, через який мережа IMS буде взаємодіяти з TM3K або GSM.

– MGCF – керує транспортними шлюзами.

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

– MRFC – управляє процесором мультимедіа ресурсів, забезпечуючи реалізацію таких послуг, як конференц-зв'язок, оповіщення, перекодування переданого сигналу.

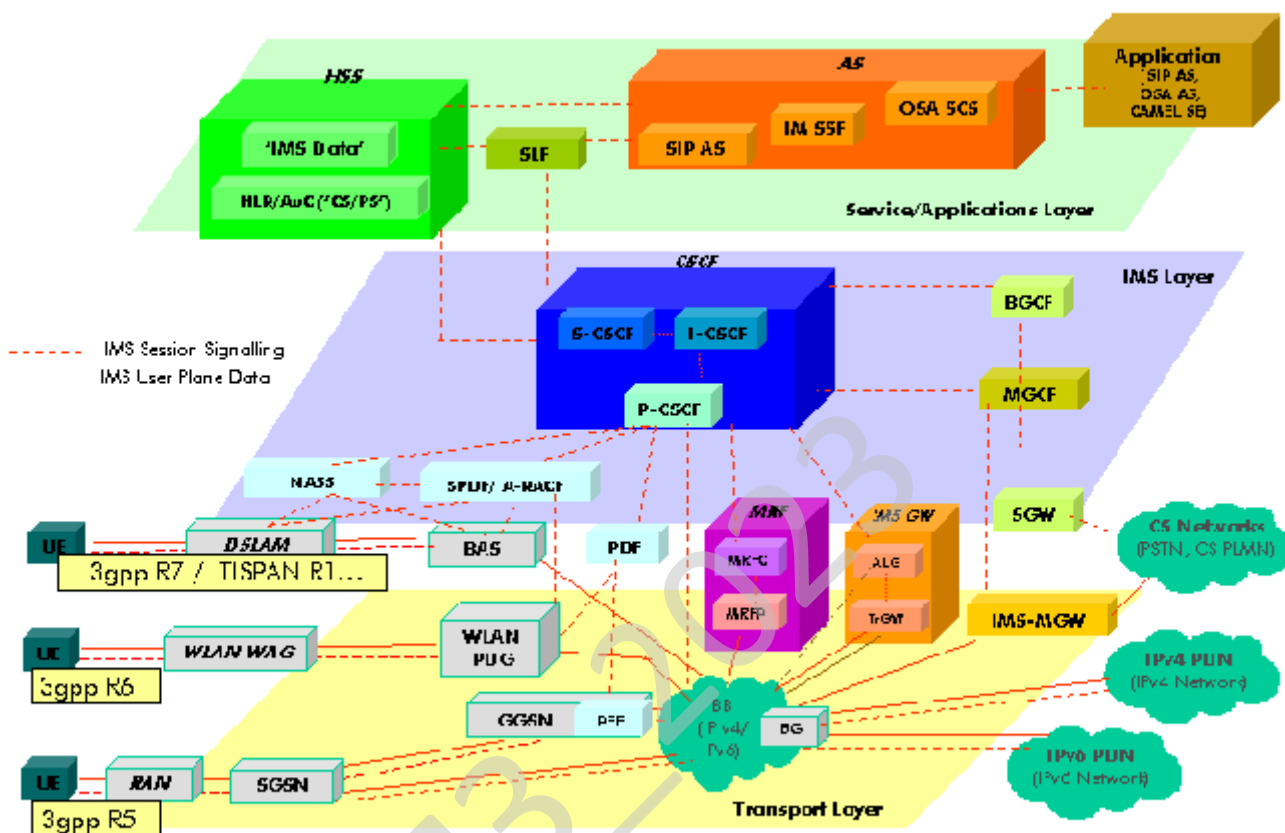


Рисунок 3.1 – Структура архітектури IMS

TISPAN

Робоча група TISPAN допрацювала архітектуру від 3GPP, додавши елементи для взаємодії з широкосмуговими мережами. Підтримка широкосмугових мереж забезпечується наступними елементами:

– NASS – підсистема підключення мережі, в основні завдання якої входить: динамічне призначення IP-адресів, автентифікація і авторизація, конфігурація мережі доступу, управління місцезнаходженням на рівні IP.

- RACS – підсистема управління ресурсами та доступом, забезпечує функції управління доступом на підставі інформації про доступні ресурсах та місцевої політики Session Admission Control, а також вхід в мережу за допомогою управління шлюзом.

- PSTN / ISDN Emulation – емуляція мережі ТМЗК / ISDN, забезпечує формування набору послуг, які, з точки зору абонента, повністю відповідають послуг мережі ТМЗК.

Один із двох основних логічних блоків IMS – “серце” всієї системи – це блок керування сеансами зв'язку (Call Session Control Function – CSCF), або, попросту говорячи, SIP-сервери. Їхнє основне завдання – обробка SIP-запитів з метою організації сеансів мультимедіазв'язку між користувачами. Вони “стежать” за виконанням правил безпеки й виділенням необхідних ресурсів для надання різних послуг. У завдання CSCF входить керування іншими мережними елементами (медіашлюзами, прикордонними пристроями й т.п.) – знов-таки для належного обслуговування користувачів. Логічно сервери керування сеансами зв'язку діляться на три групи: Serving-CSCF (S-CSCF), Proxy-CSCF (P-CSCF) і Interrogating-CSCF (I-CSCF), однак докладний розгляд того, як “розподіляються повноваження” між цими елементами, виходить за рамки даної статті.

Другий по важливості блок IMS – “мозок” системи – це абонентська база даних (Home Subscriber Server – HSS). У першому наближенні HSS можна зрівняти з використовуваним у стільникових мережах реєстром HLR, у якому зберігається інформація про активних абонентів і їхнє місцезнаходження. Однак функції HSS значно ширше. Це база даних з інформацією не тільки по абонентах мобільних мереж, але й по абонентах мереж фіксованого зв'язку (як уже говорилося вище, для IMS неважливо, яким способом підключений абонент). У ній зберігається інформація про різноманітні переваги абонента, наприклад, по переадресації й фільтрації викликів, оповіщенні й повідомленнях голосової пошти, персональна адресна книга (buddy list) для розсилання повідомлень і організації конференцій. Також на сервері HSS є всі необхідні дані для обліку

доступності/статусу (presence) і місцезнаходження (location) абонента. Замість застарілого протоколу Radius для взаємодії між HSS і серверами CSCF використовується протокол Diameter, стандартизований організацією IETF. Крім інших удосконалень, в Diameter передбачена підтримка функції тарифікації, у тому числі й для надання популярних послуг з передоплатою (prepaid).

Також показані ще два важливих елементи архітектури IMS: відповідальні за керування медіашлюзами (Breakout Gateway Control Function – BGCF, або Media Gateway Control Function – MGCF) і обробку медіапотоків (Media Resource Function – MRF). Якщо до сеансу зв'язку треба підключити абонента, що перебуває в мережі з комутацією каналів (мережа стільникового зв'язку або ТфОП), блок BGCF/MGCF забезпечує доведення до її відповідної сигнальної інформації. При необхідності він перетворить сигнальні повідомлення з формату SIP у формат ISUP. Подібна функціональність типова для комутаторів softswitch, але в архітектурі IMS вона виділена в окремий логічний елемент.

Системи MRF забезпечують обробку медіапотоків, переданих між серверами додатків і кінцевих пристроїв. Їхньої функції – програвання різних голосових повідомлень, транскодування інформаційних потоків, “змішування” мовних/відеопотоків у конференцію й т.п. Виконання цих функцій ініціюється серверами CSCF прямо або через сервери додатків. На закінчення цього розділу відзначу, що на нашій спрощеній схемі показані далеко не всі елементи IMS, але основні є, і вони коротко описані вище.

3.2 Розробка структурної схеми

На рисунку 3.2 зображено структурну схему системи. IMS (IP Multimedia Subsystem) – технологія передачі мультимедійних даних в електрозв'язку на основі протоколу IP. Сама концепція є розвиток SoftSwitch, суть полягає в можливості надання необмеженого списку послуг, розподіляючи кожен як функцію на певний логічний сервіс.

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

– S-CSCF (Serving CSCF) – центральний вузол мережі IMS, обробляє всі SIP-повідомлення, якими обмінюються кінцеві пристрої.

– HSS (Home Subscriber Server) – сервер домашніх абонентів, є базою користувальницьких даних і забезпечує доступ до індивідуальних даних користувача, пов'язаними з послугами. У випадку якщо в мережі IMS використовується кілька серверів HSS, необхідне додавання SLF (Subscriber Locator Function) який займається пошуком HSS з даними конкретного користувача.

– BGCF – елемент керуючий пересиланням викликів між доменом комутації каналів і мережею IMS. Здійснює маршрутизацію на основі телефонних номерів і вибирає шлюз у домені комутації каналів, через який мережа IMS буде взаємодіяти із ТфОП або GSM.

– MGCF – управляє транспортними шлюзами.

– MRFC – управляє процесором мультимедіа ресурсів, забезпечуючи реалізацію таких послуг, як конференц-зв'язок, оповіщення, перекодування переданого сигналу.

Архітектура IMS-системи передбачає можливість використання її елементів для надання безлічі послуг і роботи безлічі додатків. Це дозволяє скоротити як капітальні витрати на встаткування й ПЗ, так і витрати, пов'язані з їхнім обслуговуванням і технічною підтримкою. Знов-таки в наявності принципова відмінність від традиційних систем, у яких засобу керування послугами і їхніми доставками жорстко “зв'язані” з конкретною послугою (наприклад, АТМ – телефонія, сервер MCU – відео-конференц-зв'язок і т.п.), а тому впровадження принципово нового сервісу вимагає побудови відповідної інфраструктури для його доставки.

Реалізувавши принципи IMS, оператор може серйозно заощадити й при нарощуванні потужностей своєї мережі. При використанні традиційної – “монолітної” – системи оператор змушений модернізувати її повністю, навіть коли потрібно підвищити ємність (або інші характеристики) тільки одного

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

логічно виділеного функціонального блоку. В “листяній” мережі IMS кожний шар можна нарощувати окремо: транспортний – коли підвищується обсяг трафіку; керування сервісами (сеансами зв'язку) – коли росте число абонентів і/або сеансів зв'язку; нарешті, прикладний – коли росте популярність конкретного сервісу або необхідно впровадити новий. Зрозуміло, що при цьому в оператора є широкі можливості по оптимізації своїх інвестицій у нові апаратні й програмні засоби.

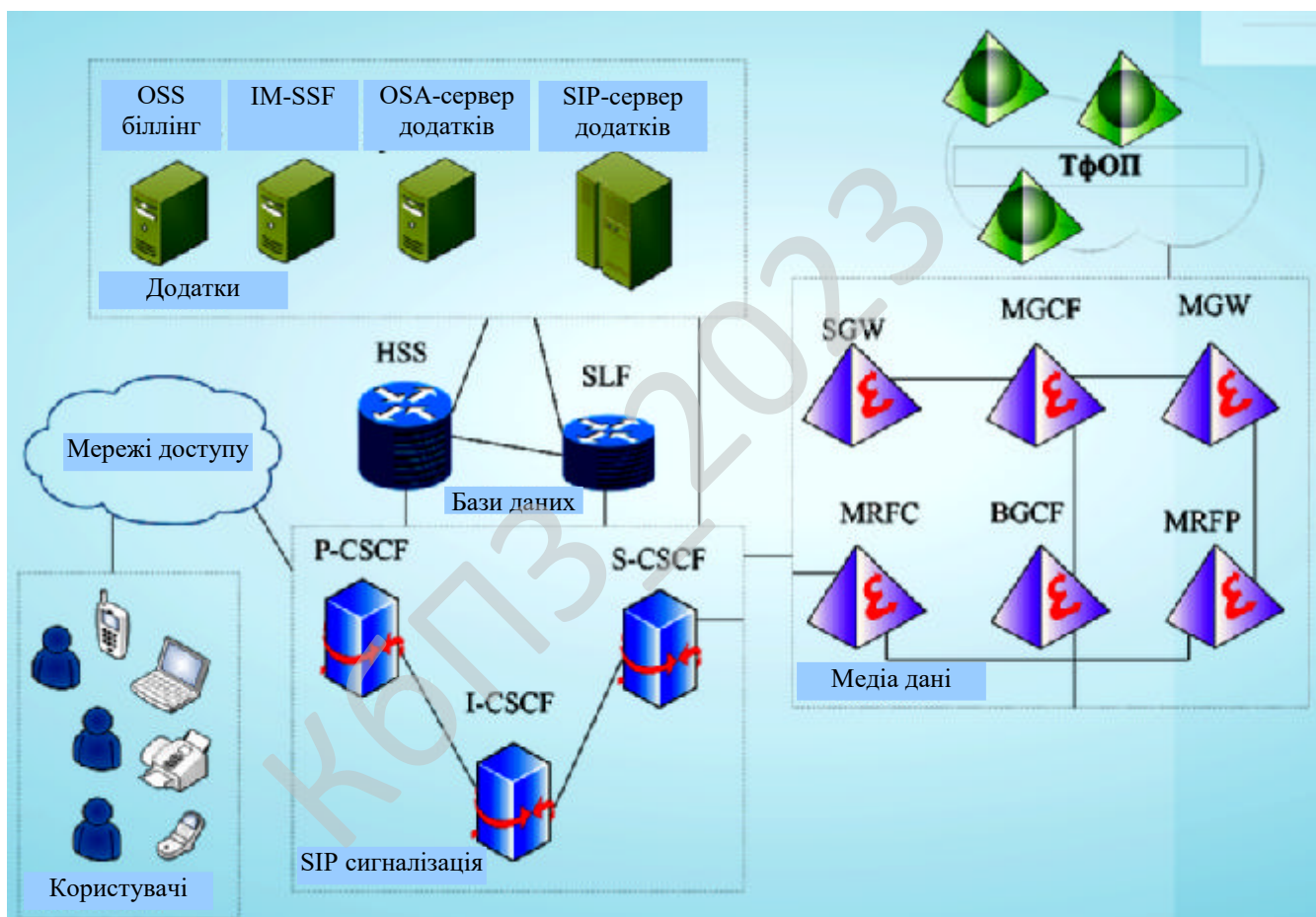


Рисунок 3.2 – Структурна схема системи

Реалізація нових послуг ще одне, уже не раз перевага, що повторювалася в статті, архітектури IMS. Незалежність IMS від специфіки мережного транспорту й каналів доступу робить її відмінною основою для конвергенції служб фіксованого й мобільного зв'язку (Fixed Mobile Convergence – FMC). Але тут

важливо помітити, що IMS аж ніяк не єдино можливий технологічний фундамент FMC. Більше того, на початковому етапі конвергенції економічно вигідними, швидше за все, виявляться інші рішення.

Проблеми

Серйозним фактором, що стримує повноцінну реалізацію архітектури IMS, є те, що переважна більшість телефонних мереж, включаючи й мережі стільникового зв'язку, ґрунтуються на комутації каналів, а “ідеологія” IMS – пакетна. Причому не просто пакетна: у документах 3GPP прописане використання протоколу IP версії 6 (IPv6). Це значить, що для повноцінного переходу до IMS операторів зв'язку прийде не тільки вибудувати нову схему керування сеансами зв'язку й модернізувати системи підтримки експлуатації (OSS) і бізнес-операцій (BSS), але й істотно попрацювати над базовою інфраструктурою, забезпечивши підтримку маршрутизаторами протоколу IPv6. Здається, далеко не всі оператори готові найближчим часом піти на такі масштабні інвестиції.

Ще одна велика проблема – відсутність терміналів, орієнтованих на роботу в IMS-мережах. Якщо в традиційних телефонних мережах термінали (телефони) – це пристрої, що володіють мінімумом “інтелектуальних” функцій, то технології IMS вимагають досить високого рівня цього самого “інтелекту” для роботи зі складними прикладними службами. Коли ми говоримо про IMS-сумісні термінали, то в першу чергу маємо на увазі, що мобільна трубка або інший абонентський пристрій постачені клієнтом, здатним ініціювати й обробляти IMS-запити. Крім того, цей клієнт повинен підтримувати роботу таких додатків, як, наприклад, push-to-talk. Ці додатки можуть поставлятися із самим терміналом або завантажуватися з мережі.

Сьогодні ситуація така, що розроблювачі термінального встаткування не готові почати масове виробництво IMS-сумісних терміналів з тієї причини, що самих мереж IMS ще дуже й дуже мало. Разом з тим відсутність терміналів (і нових додатків) серйозно гальмує розгортання нових мереж. Тільки коли це

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

- 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 середа 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

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-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

(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 у складі IMS 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-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

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 у складі IMS 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

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 у складі IMS 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 у складі IMS, 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 вид,

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

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 у складі IMS 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 OKS7 і інших видів сигналізації, а також прозора трансляція сигнальної інформації з мережі IP-телефонії у складі IMS. У мережі, побудованої на базі рекомендації H.323, сигналізація OKS7, 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 встаткування 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 у складі IMS, 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 у складі IMS сьогодні побудована на базі

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

протоколу H.323. В останньому зі згаданих підходів (у проекті 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.

3.3 Розробка функціональної схеми

На функціональній схемі, зображеній на рисунку 3.3, 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 IP 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).

Взаємодія користувача із системою здійснюється в діалоговому режимі. Основним сполучним елементом розроблювального програмного забезпечення корпоративної мережі зв'язку є система меню, що складається з головного меню та підменю. Розроблена система є меню-орієнтованою. У вікні завантаження, яке спливає при завантаженні програми, відображено інформацію про програмне забезпечення та інформацію про автора.

На головній формі існують наступні закладки:

- про програму;
- система доступу;
- системне меню програми;
- настройки програмного забезпечення;
- вихід з програми.

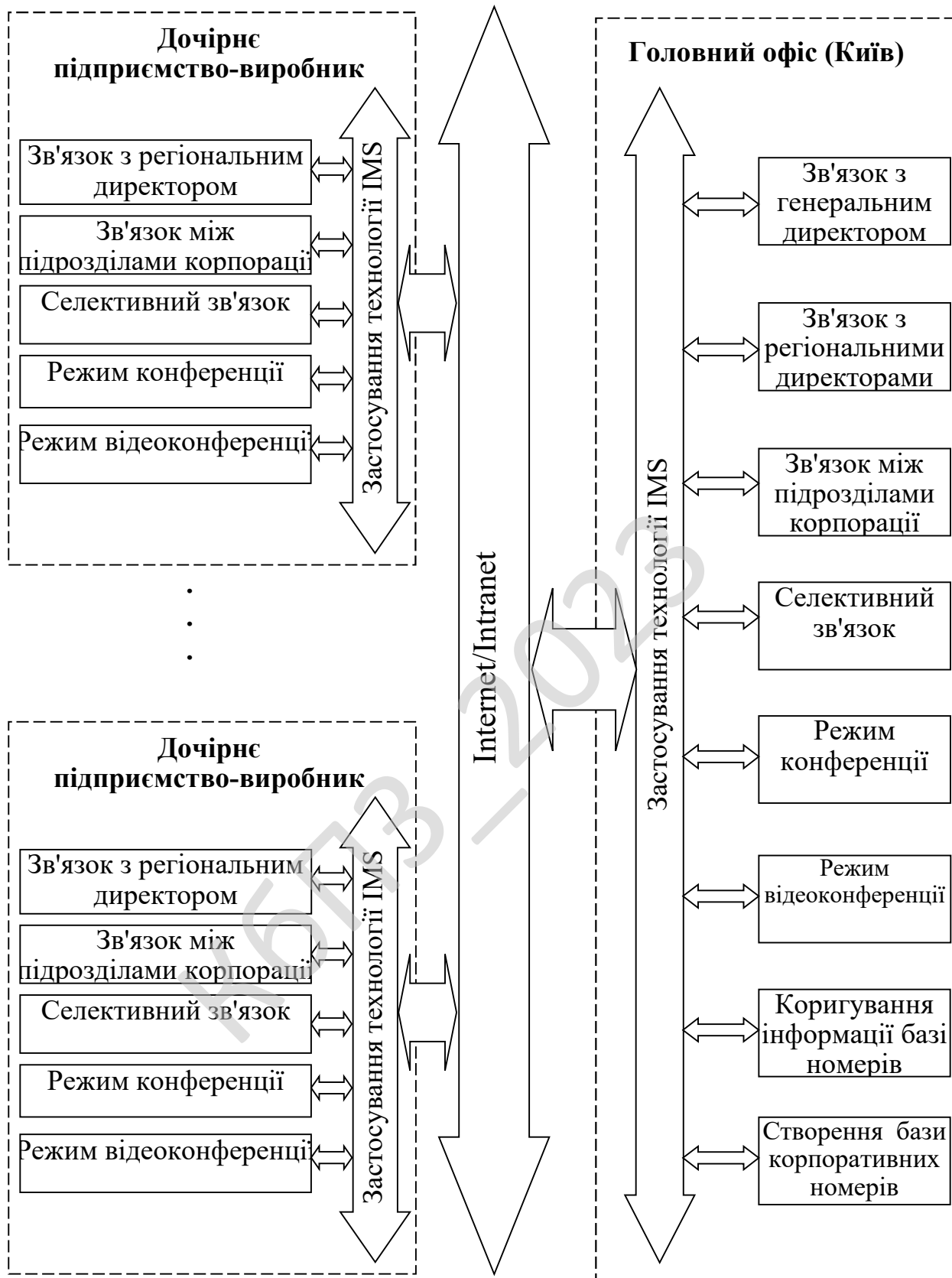


Рисунок 3.3 – Функціональна схема системи

На закладці «Про програму» міститься коротка інформація про дану програму: ПІБ розробника, місце та рік розробку, версію програми.

На закладці «Система доступу» сформована форма автентифікації користувача, на якій потрібно ввести паролльні дані, для доступу до мережі корпоративного зв'язку, що також допомагає вести моніторинг дзвінків всередині корпорації. Для підвищення захищеності корпоративної мережі зв'язку передбачено можливість «заміни пароля».

На закладці «Системне меню програми» реалізована можливість настройки зовнішнього вигляду програмного забезпечення.

На закладці «Настройки програмного забезпечення» впроваджена система створення піктограм до вигляду додатків MS-Office, які є стандартом де-факто інтерфейсного забезпечення програмних додатків.

Закладка «Вихід із програми» дозволяє користувачу закінчити роботу з програмою і вийти з додатка.

На «Головній формі» розташовані наступні програмні блоки для реалізацій функцій мережі корпоративного зв'язку усередині корпорації:

- Рядок статусу програмного забезпечення.
- Виклик IP-адресату.
- Завершення сеансу зв'язку.
- Редагування легенди IP-номерів.
- Редагування бази даних IP-адрес.
- Моніторинг бази даних IP-адрес та легенд.
- Організація селективного зв'язку.

У пункті меню «Рядок статусу програмного забезпечення» відображено поточний статус програмного забезпечення, тобто, що у даний момент виконує програма: Виклик IP-адресату, Завершення сеансу зв'язку, Редагування легенди IP-номерів, Редагування бази даних IP-адрес, Моніторинг бази даних IP-адрес та легенд або Організація селективного зв'язку. При цьому, якщо іде розмова, або селективний зв'язок то указується з якими абонентами відбувається ця дія.

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

Пункт меню «Виклик IP-адресату» дозволяє по вибраному з бази даних або набраному вручну IP-адресу установити зв'язок з відповідним адресатом.

Кнопка «Завершення сеансу зв'язку» нажимається при завершенні розмови з відповідним абонентом, або при завершенні селективної наради.

В пункті меню «Редагування легенди IP-номерів» вноситься легенда для кожної IP-адреси або телефонного номера абоненту, наприклад: 127.0.0.1 – Генеральний директор.

При заходженні до пункту меню «Редагування бази даних IP-адрес» з'являється можливість редагувати IP-адреси та телефонні номери відповідних IP-телефонів, тобто вносити нові дані та знищувати непотрібні телефонні номери.

Під час роботи з пунктом «Моніторинг бази даних IP-адрес та легенд» можливий перегляд, додавання, знищення та ранжування по важливості відповідних IP-адрес та їх легенд.

Пункт меню «Організація селективного зв'язку» дозволяє організувати селективний зв'язок, тобто користувач має можливість одночасно вести розмову з тими адресатами, яких він вибере для спілкування, наприклад це може бути проведення ранкового засідання генерального директора з регіональними директорами, або іншими відповідними особами підрозділів корпорації.

Як було обґрунтовано раніше, сучасна IP-телефонія у складі IMS 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 у складі IMS 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

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

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 у складі IMS. 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 у складі IMS базується 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 у складі IMS, 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.

Модуль керування підтримує три види сигналізації: H.225, H.245 і RAS. Цей модуль забезпечує реєстрацію терміналу у gatekeeper, установа 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.

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

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 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 зворотнє 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.

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 у складі IMS 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 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 H.323.

Рекомендація H.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

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

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 МС, то для кожної знову створюваної конференції повинна проводитися процедура визначення встаткування, для того, щоб виявити той з контролерів МС, що буде управляти даною конференцією. При організації централізованої конференції, крім контролера МС, повинен використовуватися процесор МР, що обробляє користувальницьку інформацію й відповідає за перемикання або змішування мовних потоків, відеоінформації й даних. При організації децентралізованої конференції процесор МР не використовується.

3.4 Розробка діаграми процесів

Діаграма процесів розробленої системи зображена на рисунку 3.4. Після початку роботи розробленого ПЗ ми потрапляємо до головного блоку системи звідки через ланку дій відбувається наступне:

- Інтерфейс ПЗ.
- Перевірка цілісності ПЗ.
- Налаштування ПЗ.
- Модуль захисту ПЗ
- Моніторинг даних.
- Обробник помилок.
- Перевірка ступені захищеності.
- Перевірка ступені працездатності.
- Модуль захисту.
- Виклик адресата.
- Селективний зв'язок.

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

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

використовується для візуалізації процесів обробки даних (структурне проектування).

Для розробника вважається звичним спочатку креслити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи. Ця діаграма в подальшому підлягає уточненню шляхом деталізації процесів та потоків даних з метою показати систему що розробляється.

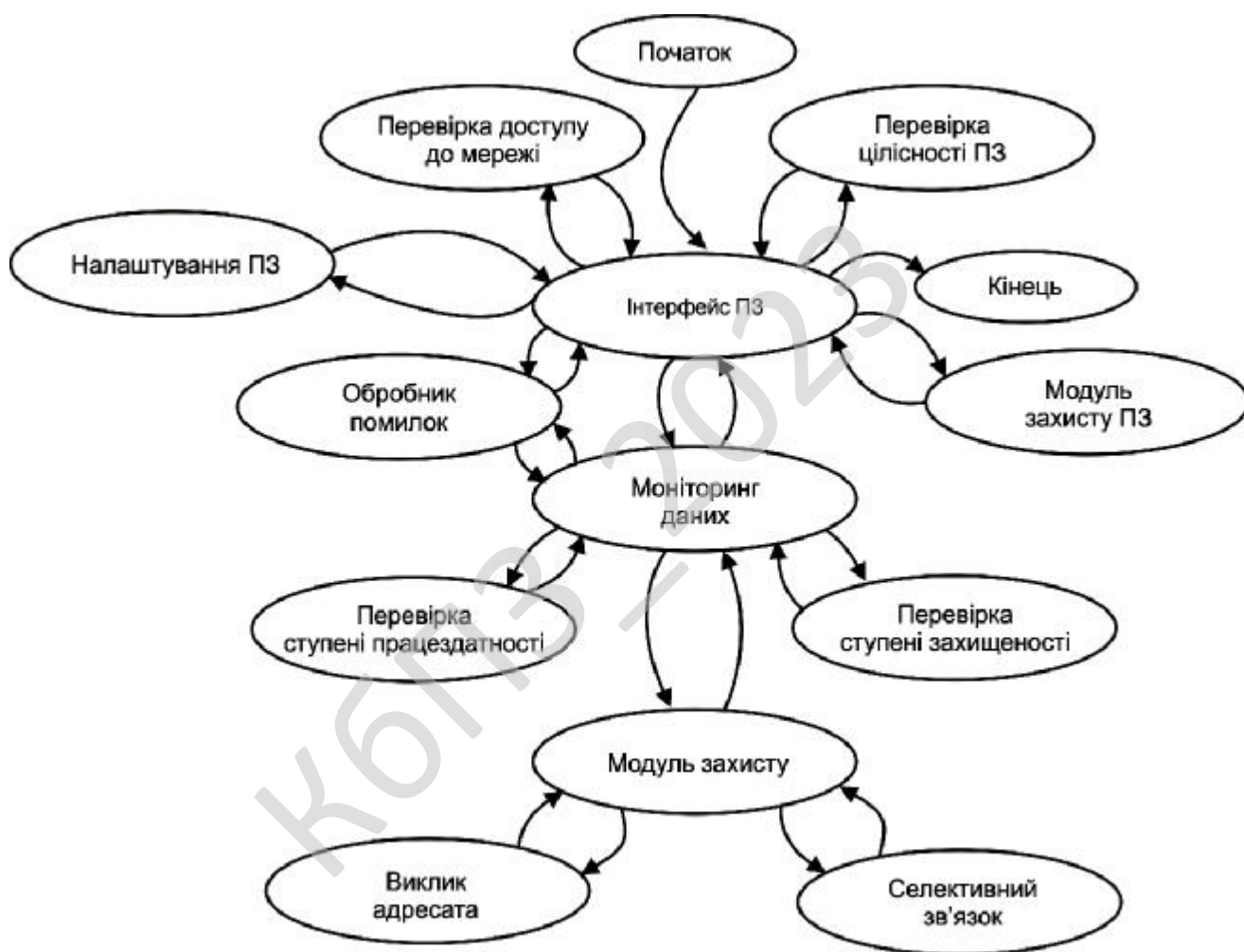


Рисунок 3.4 – Діаграма взаємодії процесів

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

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

Первинною стадією без якої не відбувається розробка програмного забезпечення це звичайно розробка блок-схем. На рисунку 4.1 зображена основна блок-схема програми, на рисунку 4.2 зображено роботу підпрограми.

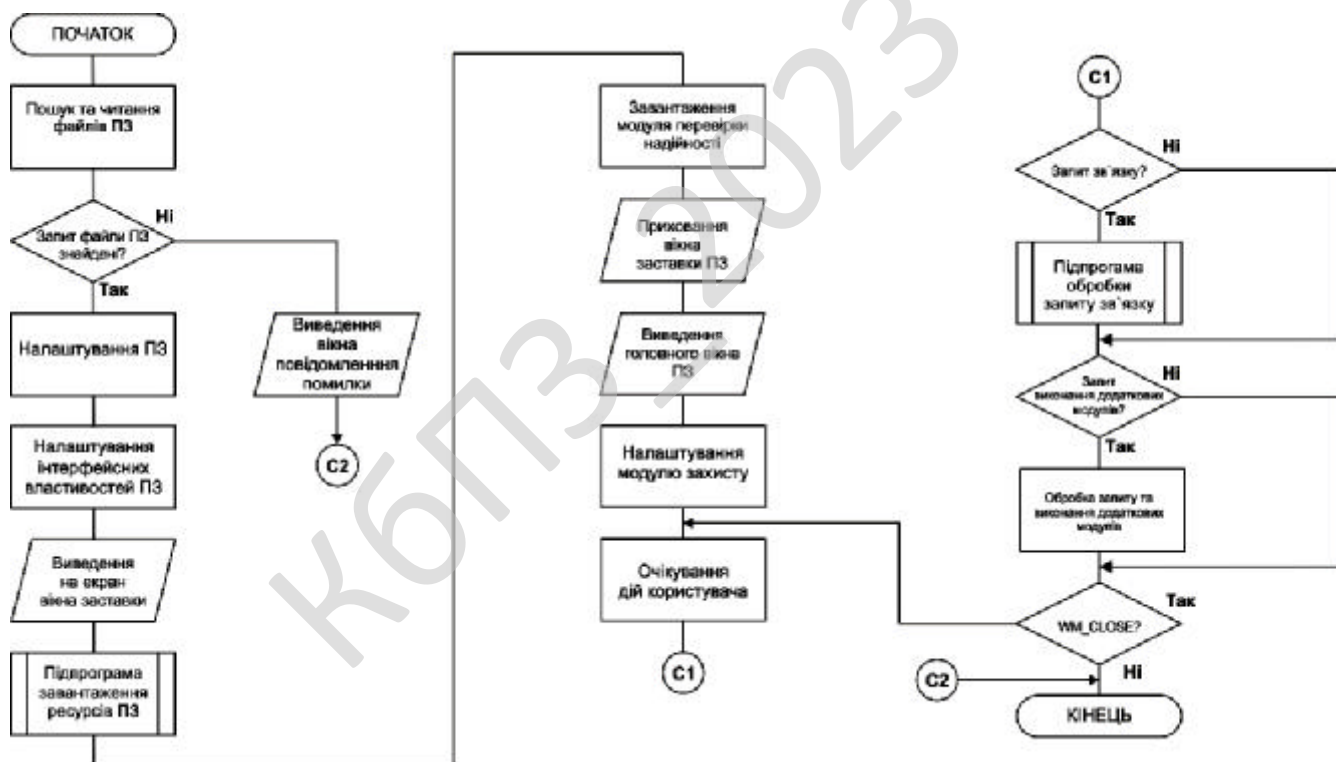


Рисунок 4.1 – Блок схема основної програми

З якої видно що робота основної програми складається з початкових етапів ініціалізації ПЗ, перевірки наявності ресурсів системи, блоку початку основного циклу з чеканням запиту від користувача в якому відбувається виклик

підпрограми та останньої стадії – перевірка поточного стану з завершенням роботи розробленого ПЗ. При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання.

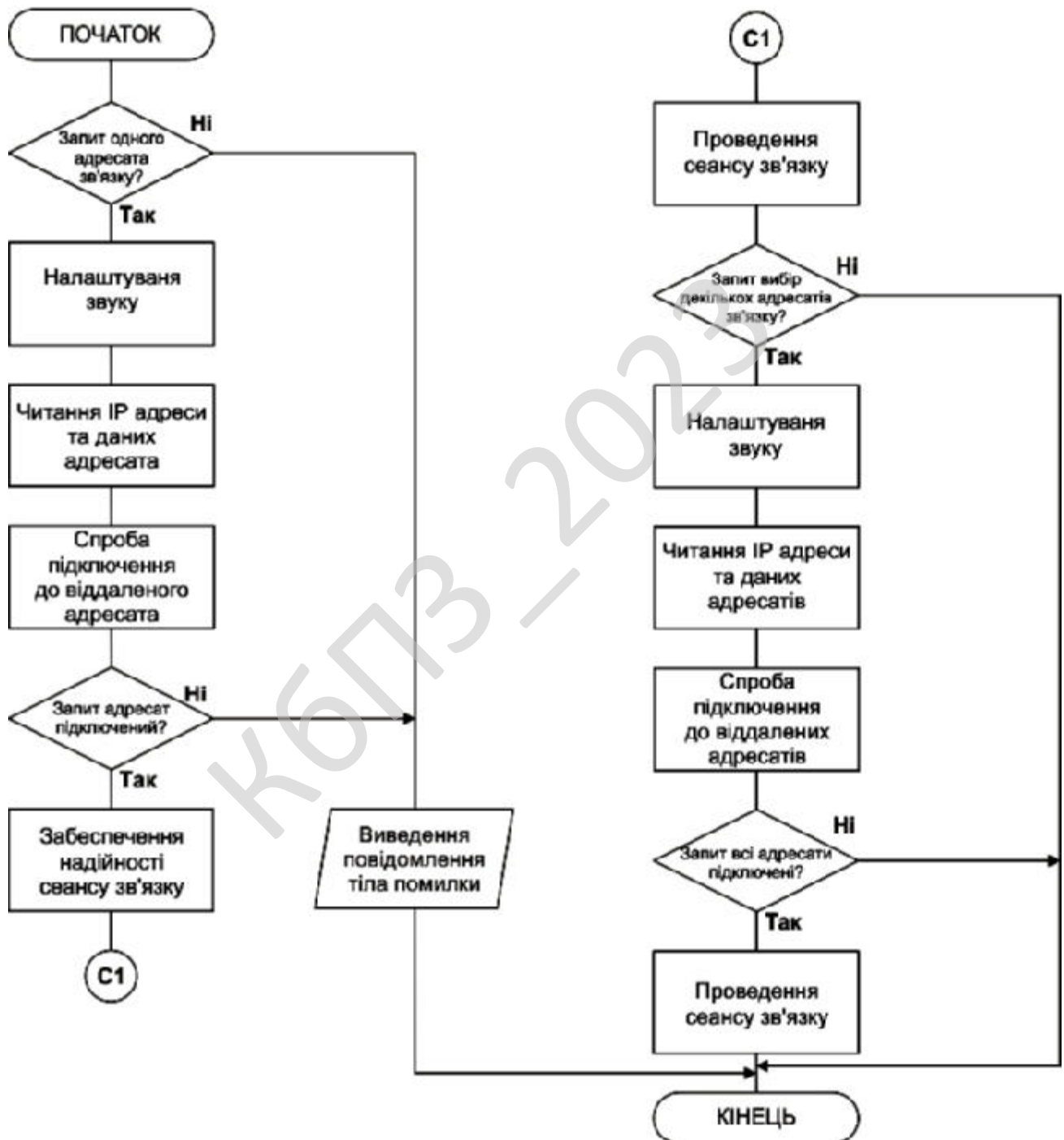


Рисунок 4.2 – Блок схема підпрограми

При складанні блок-схем програмного забезпечення і напрацювання алгоритмів я зіткнувся з масою проблем, які вимагали напрацювання процедур і функцій над основною проблематикою. Для чого були створені додаткові класи, типи даних і константи, що забезпечило вирішення проблем.

Опис алгоритмів функціонування системи

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 магістерському проекті корпоративної системи зв'язку з використанням IMS-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.

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 відрізків 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 IMS-мережах – у терміналі користувача (комп'ютері або IMS-телефоні).

Процес аналого-цифрового перетворення одержав, стосовно до систем зв'язку, назву імпульсно-кодової модуляції (ІКМ).

Щоб знизити необхідну швидкість передачі біт, застосовують нелінійний (логарифмічний) 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 A -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 A -Закону. Тому при міжнародному зв'язку в багатьох випадках потрібне перетворення m -закону в A -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.

Щоб зменшити властиву ІКМ надмірність і знизити вимоги до смуги

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

пропущення, 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 пропущення.

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

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

безпосередньо, кодують різницю між вхідним сигналом і «передвщеною» 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.

Перелік вихідних даних

У ході розробки корпоративної системи зв'язку з використанням IMS-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 IMS-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 IMS-телефонії повинні бути реалізовані багато інших функцій, спосіб реалізації яких не є об'єктом стандартизації. На передавальній стороні устаткування IMS-телефонії працює за принципом

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

«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 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. 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. 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 IMS-телефонії. У випадках, коли система, взаємодіюча 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

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

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 IMS-телефонії.

– 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;

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

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 IMS-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 IMS-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 IMS-телефонії.

Розглянемо код основної процедури.

```
program IMS;  
// Назва програми  
Uses  
// підключення модулів (стандартні)
```

```

Forms,
Dialogs,
Windows,
Sysutils,
Messages,
// підключення модулів (користувача)
  useimple in 'useimple.pas' {Form1},
  Unit2 in 'Unit2.pas' {Form2},
  Unit3 in 'Unit3.pas' {Form3},
  Unit1 in 'Unit1.pas' {Form0},
  Unit4 in 'Unit4.pas' {Form4};
// підключення ресурсів
{$R simple.RES}
Var
// створення змінних
  i:integer;
// функція захисту
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;
// початок роботи
begin
Application.Initialize;
// ініціалізація
if FileExists('main.dat')=false then
begin
// Якщо файл main.dat не знайдено завершуємо роботу
MessageDlg('Файл main.dat не знайдено! завершення ПЗ',mtInformation,[mbOK],0);
Application.Terminate;
end else
begin
// підключення файлу main.dat
AssignFile(F, 'main.dat');  Reset(F);  Readln(F, S);  CloseFile(F);
// Автентифікація

```

						ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			65

```

if Crypt(TextBox('Увага!', 'Введіть пароль', Y), KEY, false) = S then
begin
    MessageDlg('Дякую, пароль вірний!', mtInformation, [mbOK], 0);
    Try
// створення форми (динамічне)
        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 := 'IMS';
// завантаження форм
        Application.CreateForm(TForm1, Form1);
        Application.CreateForm(TForm3, Form3);
        Application.CreateForm(TForm0, Form0);
        Application.CreateForm(TForm4, Form4);
        Finally
            Form2.Free;
// звільнення динамічної форми
        End;
        Application.Run;
// запуск ПЗ
        end
        else
        begin
            MessageDlg('Невірний пароль!', mtInformation, [mbOK], 0);
            Application.Terminate;
        end;
    end;
end.

```

Опис математичного забезпечення

Для передачі мови по IMS-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

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

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 IMS-адресам й формування відповідної легенди та власне сама робота програми у режимі IMS-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 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 IMS-технології.

Програма складається з наступних основних модулів.

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

Процедура обробки головного меню – запуск відповідної процедури. Процедура введення даних – забезпечення введення інформації у бази даних IMS-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 IMS-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 запиту 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.

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

4.2 Захист розробленого програмного забезпечення

Захист розробленого програмного забезпечення буде відбуватися за допомогою Serpent – симетричний блочний алгоритм шифрування, розроблений Россом Андерсоном, Елі Біхамом та Ларсом Кнудсенем. Алгоритм був одним з фіналістів 2-го етапу конкурсу AES. Як і інші алгоритми, які брали участь у

конкурсі AES, Serpent має розмір блоку 128 біт і можливі довжини ключа 128, 192 або 256 біт. Алгоритм являє собою 32-раундовий шифр на основі SP-мережі, і працює з блоком з чотирьох 32-бітових слів. Serpent був розроблений так, що всі операції можуть бути виконані паралельно, використовуючи 32-а 1-бітних «потоків».

При розробці Serpent використовувався консервативніший підхід до безпеки, ніж у інших фіналістів AES, проектувальники шифру вважали, що 16 раундів достатньо, щоб протистояти відомим видам криптоаналізу, але збільшили число раундів до 32, щоб алгоритм міг краще протистояти ще не відомим методам криптоаналізу.

Шифр Serpent не запатентований і є громадським надбанням.

Алгоритм створювався під гаслом «криптографічний алгоритм 21 століття» для участі в конкурсі AES. При створенні нового алгоритму Serpent його автори дотримувалися консервативних поглядів на проектування, що підтверджується первісним рішенням про використання таблиць підстановки з відомого багато років раніше алгоритму шифрування DES, який протягом довгого часу вивчався провідними фахівцями в області криптографії та захисту інформації і чий властивості і особливості були добре відомі науковому світу. Одночасно з цим до нового алгоритму міг бути застосований вичерпний аналіз, вже розроблений для DES. Не використовувалися нові, неперевірені і невикробувані технології при створенні шифру, який у разі прийняття був би використаний для захисту величезних масивів фінансових транзакцій та урядової інформації. Основною вимогою до учасників конкурсу AES було те, що алгоритм-претендент повинен бути швидшим, ніж 3DES, і надавати як мінімум такий же рівень безпеки: він повинен мати блок даних довжиною 128 біт і ключ завдовжки 256 біт. 16-раундовий Serpent був би таким же надійним, як 3DES, але в два рази швидшим. Однак, автори вирішили, що для більшої надійності варто збільшити кількість раундів до 32. Це зробило шифр таким же швидким, як DES, і набагато надійнішим, ніж 3DES.

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

використовуваного ключа може приймати різні значення, але для конкретики зафіксуємо їх довжину в 128, 192 або 256 біт. Короткі ключі довжиною менше 256 біт доповнюються до повної довжини в 256 біт.

Шифрування складається з наступних основних кроків:

- Початкова перестановка.
- 32 раунд, кожен з яких складається з операції змішування з 128-бітовим ключем (побітове логічне виключаюче «або»), таблична заміна (S-box) і лінійне перетворення. В останньому раунді лінійне перетворення замінюється додатковим накладанням ключа.

- Кінцева перестановка.

Початкова і кінцева перестановки не мають будь-якої криптографічного значущості. Вони використовуються для спрощення оптимізованої реалізації алгоритму і підвищення обчислювальної ефективності.

Розширення ключа

Як і інші алгоритми, що брали участь в конкурсі AES, Serpent має можливі довжини ключа 128, 192 або 256 біт. «Неповний» ключ довжиною менше 256 біт доповнюється за наступним правилом: додається одиничний біт справа, за ним слід стільки нульових бітів, щоб довжина ключа стала дорівнює 256 бітам.

Початкова перестановка IP

Дана перестановка IP задається таблицею, де вказується позиція, на яку перейде відповідний біт (наприклад, біт 1 перейде на 32 позицію):

S-бокси (таблиці замін)

В алгоритмі Serpent таблиці замін є 4-бітовими перестановками з властивостями стійкості до диференціального криптоаналізу, до лінійного криптоаналізу і такою властивістю, що порядок вихідних біт, як функції вхідних повинен бути максимальний, тобто бути рівним 3.

Таблиця підстановки генерується з відомих і добре вивчених таблиць для алгоритму DES в ітераційному процесі, поки не будуть отримані бажані

диференціальні й лінійні властивості. Таким чином, створюється 8 таблиць підстановки.

Лінійне перетворення LT

Лінійне перетворення LT задається таблицею, де біти перераховані від 0 до 127 (наприклад, вихідний 2 біт утворений 2, 9, 15, 30, 76, 84, 126 битами, складеними за модулем 2). В кожному рядку описується 4 вихідних біти, які разом складають вхідні дані на одну таблицю замін в наступному раунді. Варто зазначити, що даний набір являє собою таблицю $IP(LT(FP(x)))$, де LT і є те лінійне перетворення.

Таблиця зворотного лінійного перетворення, яке використовується при розшифровці ILT.

Кінцева перестановка FP

Дана перестановка є зворотною до початкової, тобто $FP=IP^{-1}$ і задається наступною таблицею.

Ефективна реалізація алгоритму

Бажання авторів зробити алгоритм саме таким, яким він є стає зрозумілим при розгляді його ефективної низькорівневої реалізації.

Serpent був створений таким чином, щоб всі операції в процесі шифрування і розшифрування одного блоку могли бути виконані паралельно в 32 потоках. До того ж низькорівневий опис алгоритму набагато простіший, ніж стандартний опис. Ніяких початкових і кінцевих перестановок не потрібно.

Шифрування складається з 32 раундів. Відкритий текст є першими проміжними даними $V_0 = P$. Потім виконується 32 раунди, кожен i -й раунд складається з:

– Змішування з ключем. Проводиться побітове виключаюче «або» проміжних даних V_i з ключем довжиною 128 біт.

– Застосування таблиць підстановки. Вхідні дані довжиною 128 біт поділяються на 4 слова по 32 біта. Таблиця підстановки, реалізована послідовністю логічних операцій (як якщо це було б реалізовано апаратно),

застосовується до цих 4 слів. В результаті виходить 4 вихідних слова. Таким чином, центральний процесор виконує підстановку по 32 копії таблиці одночасно.

– Лінійне перетворення. 32-бітові слова перетворюються заданим порядком.

Першою причиною вибору такого лінійного перетворення є максимізація лавинного ефекту. Такі таблиці підстановки мають властивість, що зміна кожного вхідного біта призведе до зміни 2 вихідних бітів. Таким чином, кожен вхідний біт відкритого тексту вже через 3 раунди впливає на всі вихідні біти. Аналогічно кожен біт ключа впливає на результат шифрування.

Друга причина полягає в простоті перетворення. Воно може бути реалізоване на будь-якому сучасному процесорі з мінімальними витратами.

КБПЗ_2023

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розглянемо розроблене ПЗ яке зображено на рисунку 5.1. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні розділи:

- Розділ меню: Сеанс зв'язку; Налаштування.
- Розділ вкладок: Виклик персоналу; Інструкція користувача; Про програму.
- Розділ нижнього інформаційного меню статусу.

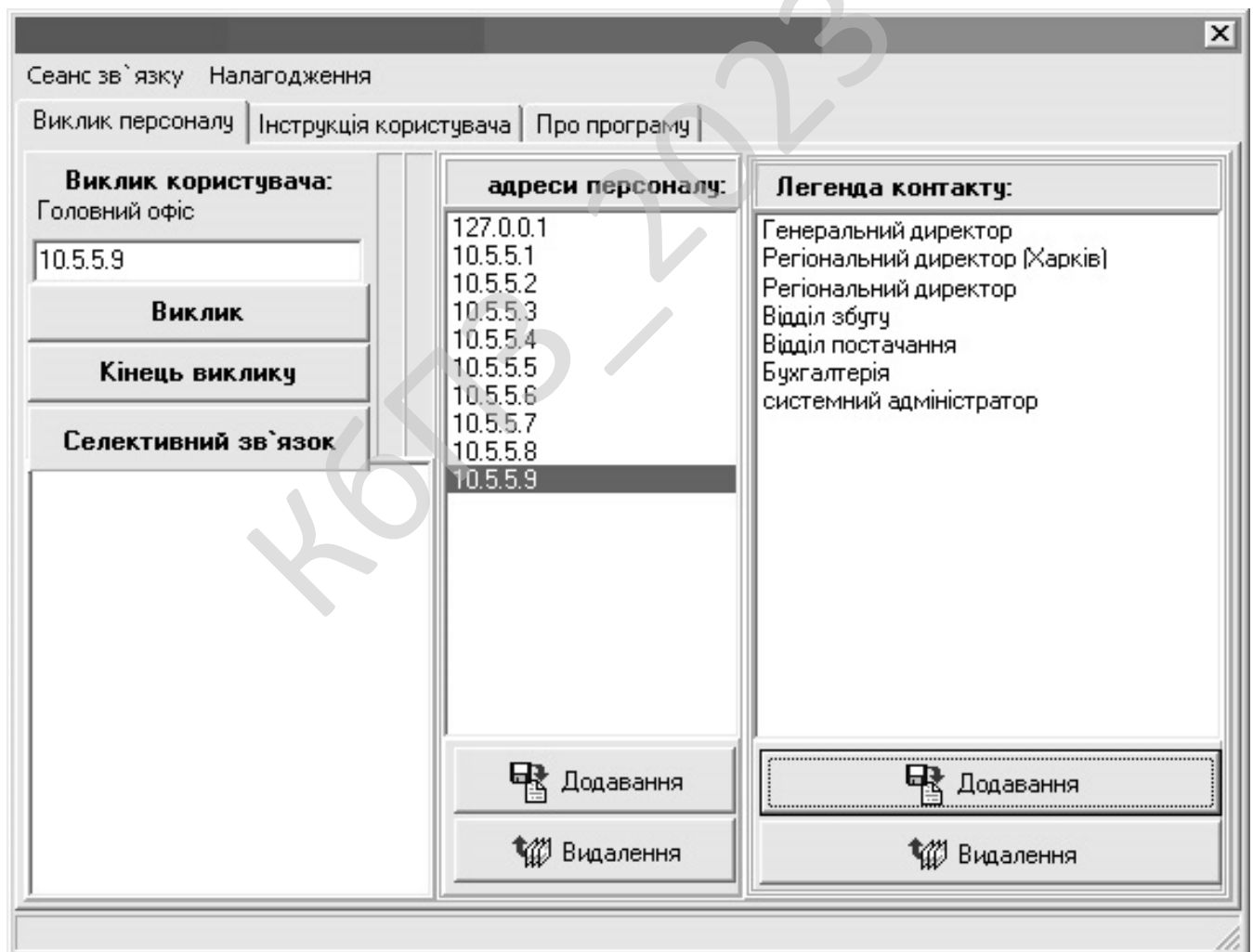


Рисунок 5.1 – Головне вікно ПЗ

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

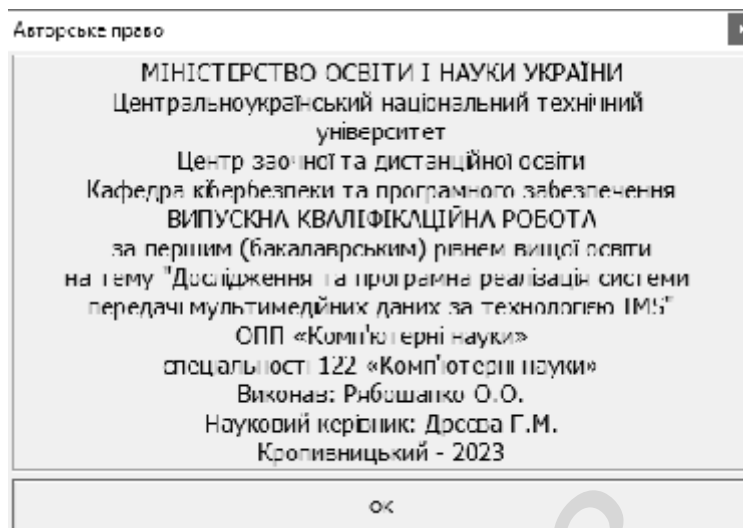


Рисунок 5.2 – Авторське право

Обрано умови розповсюдження – Shareware. Під умовно-безплатним програмним забезпеченням можна розуміти спосіб або метод розповсюдження комерційного ПЗ на ринку (тобто на шляху до кінцевого користувача), при якому випробувачеві пропонується обмежена за можливостями (не повнофункціональна або демонстраційна версія), терміном дії (тріал версія) або версія з вбудованим набридливим нагадуванням про необхідність оплати використання програми. В угоді про використання (ліцензії для кінцевого користувача, EULA) також може бути обумовлена заборона на комерційне або професійне (не тестове) її використання. Основний принцип умовно-безплатного ПЗ – «спробуй, перш ніж купити» (try before you buy). ПЗ що поширюється як умовно-безплатний, надається користувачам безоплатно. Протягом певного терміну, що становить зазвичай тридцять днів, він може користуватися програмою, тестувати її, освоювати її можливості.

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи передачі мультимедійних даних за технологією IMS.

Метою розробки є дослідження та програмна реалізація системи передачі мультимедійних даних за технологією IMS.

Об'єктом дослідження є процес передачі мультимедійних даних за технологією IMS.

Предметом дослідження є методи передачі мультимедійних даних за технологією IMS.

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

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

– Удосконалено метод передачі мультимедійних даних за технологією IMS.

– Розроблено вітчизняний продукт передачі мультимедійних даних за технологією IMS, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

7 ДАНІ ПРО ЕКОНОМІЧНУ ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ

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

Після ознайомлення з підприємством та засобами розробки програмної продукції був розроблений план розробки програми. Був підрахований необхідний час для розробки та впровадження програми. Цей час склав 48 днів (два місяці). В магістерській роботі проведене дослідження та виконана програмна реалізація системи передачі мультимедійних даних за технологією IMS. 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	57
3. Запланований термін розробки, днів	Frq	48 (2 місяці)
4. Група задачі підсистеми управління (1-6)	–	1
5. Ступінь новизни задачі (А, Б, В, Г)	–	Б
6. Складність алгоритму (1, 2, 3)	–	2
7. Кількість макетів вхідної інформації	–	3

Продовження таблиці 7.1

1	2	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

Продовження таблиці 7.1

1	2	3
28. Досвід роботи з мовою і інструментами середовища розробки (1-6)	–	2
29. Досвід роботи з програмними інструментами розробки (1-6)	–	3
30. Розробка ПО для декількох серверів одночасно (1-6)	–	2
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн	–	60000
33. Норматив додаткової зарплати, % :	Нд	10
34. Норматив відрахувань у соціальні фонди, %	Нс	22
35. Норматив загальногосподарських витрат, %	Нг	15
36. Норматив витрат на освоєння нових мов програмування, %	Нп	15
37. Рівень рентабельності програмної продукції, %	Ре	50
38. Ставка податку на додану вартість, %	Ндв	20

7.2 Розрахунок трудомісткості розробки програмної продукції

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

Визначимо трудомісткість розробки ПЗ для стадії РП.

Обчислюємо номінальні трудовитрати, люд-міс.:

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

де А – коефіцієнт Боема, А=2,45;

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

B – показник ступеня, що визначається співвідношенням

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

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

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

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

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

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

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:

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

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)%

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

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

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

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

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

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

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

$$Ч = \frac{209 \cdot 1}{48 - 5} = 4,8 \text{ ставки}$$

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

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

Виходячи з цього фонд робочого часу інженерів-електронщиків складає:

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

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

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83

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

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

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 2019, серверу доступу ADSL (OC Linux), налаштування ADSL, VPN PPPoE, Frame Relay, Wi-Fi	0,8	0,2
	Налаштування і конфігурування базової станції безпроводного зв'язку (СМТS)	0,2	
	Розробка та впровадження проектів з організації зв'язку між віддаленими об'єктами, ЛОМ	0,2	
	Забезпечення цілодобової роботи зв'язку клієнтів до мережі Інтернет	0,4	
Всього		1,6	

Продовження таблиці 7.4

Посада	Вид роботи	Час	Кількість штатних одиниць
Продакт-менеджер	Презентації нової продукції, пошук каналів збуту	2	0,5
	Підтримка постійних клієнтів	1	
	Оформлення договорів, ведення тендерів	0,5	
	Контроль взаєморозрахунків з постачальниками	0,5	
Всього		4	
Дизайнер WEB	Розробка концепції оформлення та інтерфейсу сайту, оптимізація дизайну існуючих, проектує їх структуру та навігацію	0,5	0,2
	Створення графічних і стилістичних елементів сайту	0,5	
	Оформлення банерів і промо-сторінок	0,3	
	Розміщення графіки і контенту на Інтернет сторінках	0,3	
Всього		1,6	
Інженер верстальник	Розробка та верстка макетів рекламної продукції та технічної документації	1	0,2
	Верстка друкованих видань	0,2	
	Додрукова підготовка макетів	0,2	
	Розміщення графіки і контенту на Інтернет сторінках	0,2	
Всього		1,6	

Складемо штатний розклад виконавців у таблицю 7.5.

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньо-місячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	1	15000	30000
Продакт-менеджер	0,5	12000	12000
Інженер-програміст	4,8	14000	134400
Інженер-електронщик	1,2	9000	21600
Інженер-системотехнік	0,2	9000	3600
Адміністратор мережі	0,2	11000	4400
Системний програміст	0,2	9000	3600
Дизайнер WEB	0,2	9000	3600
Інженер-верстальник	0,2	9000	3600
Бухгалтер-економіст	0,2	10000	4000
Всього за період розробки	$R_{cn}=8,7$	-	$\Phi_{роб}=220800$

Розрахуємо середньоденну зарплату одного виконавця:

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

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

$$z_{cd} = \frac{220800}{8,7 \cdot 48} = 529 \text{ грн.}$$

7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника

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

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		86

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

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

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

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

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

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

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

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

де Π_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 комерційною пропозицією фірми Brain за 10.11.23 – джерело <http://brain.com.ua/>

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		87

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Персональний комп'ютер		11771
Системний блок		7771
Процесор	AMD Ryzen 3 2200G PRO (YD220BC5M4MFB) AM4, 4 ядра, 4 потоки, 3.5 GHz, 3.7 GHz, TDP – 65 Вт, 14nm, Radeon Vega 8	-
Системна плата	GIGABYTE A520M H сокет – AM4, 5100 MHz, LAN – 1 Гбіт/с, DVI, HDMI, 1 x M.2 2280, 4 x SATA 6.0 Gb/s, Micro-ATX	-
Жорсткий диск	SSD M.2 2280 240GB Apacer (AP240GAST280-1) 240 GB, TLC, M.2, SATA III (6Gb/s)	-
Оперативна пам'ять	DDR4 8GB 2400 MHz Patriot	-
Система охолодження	ID-Cooling DK-15 PWM INTEL, AMD	-
Корпус	Logicpower 8702 – 550w 12cm	-
Кардрідер внутрішній	Transcend TS-RDF8K USB 3.0	-
інше	Клавіатура, мишка	-
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.

Для визначення необхідної кількості капітальних вкладень складемо таблицю 7.8.

Таблиця 7.7 – Балансова вартість обчислювальної техніки

Найменування обчислювальної техніки	Кількість, шт.	Ціна за одиницю, грн.	Витрати на транспортування, монтаж та випробування.	Загальна вартість, грн.
Персональні комп'ютери	8	11771	9416,8	103584,8
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Сканери	1	2800	280	3080
Копіюв. апарат	1	5965	596,5	6561,5
Всього	–	–	–	125216,3

Таблиця 7.8 – Вартість основних фондів та амортизаційні відрахування розробника

Групи та види основних фондів	Балансова вартість, грн.	Амортизація	
		Норма, %	Відрахування, грн.
1	2	3	4
Група 3			
1. Будівлі	1280000	-	-
2. Передавальні пристрої	128000	-	-
Всього по групі	1408000	5	70400

Продовження таблиці 7.8

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

Примітка: вартість автомобіля Sens (Standard+) взята по даним з автосалону «Кіровоград-Авто», джерело <http://kirovograd-avto.ukravto.ua/catalog/tm-19/model-81/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.

$$Z_o = 529 \cdot 209 / 57 = 1940 \text{ грн}$$

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

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

де H_q – норматив додаткової зарплати, %

$$Z_d = 1940 \cdot 10 \cdot 0,01 = 194 \text{ грн}$$

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

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

де H_c – відрахування на соціальні потреби, %

$$C_{oc} = 0,01 \cdot 22(1940 + 194) = 470 \text{ грн}$$

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, %

$$G_{ocn} = 1940 \cdot 15 \cdot 0,01 = 291 \text{ грн}$$

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

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

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

Згідно виданих норм приймаємо 1/6 пачку паперу на період розробки. Тоді, враховуючи, що вартість пачки паперу складає $C_n = 240$ грн., визначаємо вартість паперу за період розробки $N_M = 2$ міс:

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

$$Z_{M1} = 240 \cdot 1/6 \cdot 2 = 80 \text{ грн.}$$

Згідно виданих норм до вартості запам'ятовуючих пристроїв входить вартість CD/DVD дисків в кількості 20 примірників:

					БКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

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

де C_{∂} – вартість дисків CD/DVD: CDR TDK 700Mb, 80Min, 52x Cake box – 23,3 грн/шт., DVD-R LG 4,7Gb, 16x speed Cake box – 23,3 грн/шт.

$$Z_{M2} = 23,15 \cdot 20 = 466 \text{ грн.}$$

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

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

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.

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

$$Z_M = (80 + 466 + 1702) / 57 = 39 \text{ грн.}$$

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, %

$$O_n = 1940 \cdot 15 \cdot 0,01 = 291 \text{ грн}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_e = 57$ прим.)

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

де A_p – загальна річна сума амортизаційних відрахувань, грн.

$$A_m = 174246 \cdot 2 / (57 \cdot 12) = 509 \text{ грн}$$

Повна собівартість ПЗ визначається як сума витрат за попередніми статтями калькуляції

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

$$C_n = 1940 + 194 + 470 + 291 + 39 + 291 + 509 = 3734 \text{ грн.}$$

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

Визначимо плановий прибуток за рівнем рентабельності (P_p) 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_c – рівень рентабельності, %

$$P_p = 0,01 \cdot 50 \cdot 3734 = 1867 \text{ грн.}$$

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

Таблиця 7.9 – Нормативна калькуляція собівартості розробки програмного забезпечення задачі

Найменування статей витрат	Позначення	Величина, грн.
1	2	3
1. Основна зарплата виконавців	Z_o	1940
2. Додаткова зарплата виконавців	Z_d	194
3. Відрахування на соціальні потреби	C_{oc}	470
4. Загальногосподарські витрати	G_{ocn}	291
5. Витрати на матеріали	Z_M	39
6. Освоєння нових операційних систем, мов програмування	O_n	291
7. Амортизація основних фондів	A_m	509
8. Повна собівартість програмного забезпечення	C_n	3734
9. Плановий прибуток	P_p	1867
10. Ціна підприємства $C_n = C_n + P_p$	C_n	5601
11. Податок на додану вартість $ПДВ = 0.01 \cdot H_{об} \cdot C_n$	$ПДВ$	1120,2
12. Відпускна ціна програмної продукції $Ц = Ц_n + ПДВ$	$Ц$	6721,2

Витрати на технічне обслуговування:

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

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

Z_z – заробітна плата обслуговуючого персоналу, грн/год

Після купівлі нового програмного забезпечення кількість профілактичних годин робіт зменшилася з 200 годин на рік до 120 годин на рік, тому витрати на технічне обслуговування зменшилися з

$$Z_{p \text{ баз}} = 200 \cdot 100 \cdot 1,1 \cdot 1,22 = 26840 \text{ грн.}$$

до

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

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

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

$$Z_{ел \text{ баз}} = 0,2 \cdot 7200 \cdot 3,8 = 5472 \text{ грн}$$

$$Z_{ел \text{ нов}} = 0,2 \cdot 6600 \cdot 3,8 = 5016 \text{ грн}$$

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

Таблиця 7.12 – Розрахунок амортизаційних відрахувань

Групи основних фондів	Норма амортизації ї %	Балансова вартість, грн., за варіантами		Сума відрахувань, грн за варіантами	
		Базовий	Новий	Базовий	Новий
Програмна продукція	50	–	6721	–	3360,5
Всього відрахувань	-	–	6721	–	3360,5

$$T_{cn} = \frac{6721}{32312 - 24481} = 0,86 \text{ років}$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1. Кількість екземплярів програми	Прим.	57
2. Повна собівартість розробленої програми	Грн.	3774
3. Ціна розробленої програми	Грн.	5601
4. Плановий прибуток від реалізації розробленої програми	Грн.	1827
5. Рентабельність програмної продукції	%	50
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	1769406
7. Загальний прибуток від реалізації програмної продукції	Грн.	104139
8. Величина економічного ефекту при виготовленні програмної продукції	Грн.	77378
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років	2,8
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	6721
11. Величина економічного ефекту у користувача програмної продукції	Грн.	4471
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	0,86

7.9 Висновки

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

КБПЗ_2023

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		98

– Соціальна взаємодія – це процес спілкування між людьми у робочому колективі або через мережеві сервіси. Соціальна взаємодія вимагає дотримання правил етикету, поваги, толерантності, співробітництва та конструктивного діалогу. Соціальна взаємодія допомагає покращити настрій, мотивацію, комунікацію та творчий потенціал працівників.

Правила охорони праці і здоров'я для програмістів:

Регулярно роби перерви в роботі. Вставай із-за столу і розминай м'язи.

Налаштуй яскравість і контрастність монітору так, щоб не напружувати очі.

Використовуй ергономічну мишку і клавіатуру, які зручно лягають у руку і не викликають болю.

Слідкуй за своєю поставою. Сиди прямо і не нахилийся до екрану.

Захищай свій комп'ютер від вірусів, шпигунських програм і хакерів. Оновлюй антивірусне програмне забезпечення і не відкривай підозрілі файли і посилання.

Не забувай про соціальну взаємодію. Спілкуйся з колегами, друзями і родиною. – Відвідувай заходи, які тебе цікавлять. Не ізолюй себе від світу.

Люби свою професію, але не забувай про інші сфери життя. Розвивай свої захоплення, хоббі і таланти. Знаходь рівновагу між роботою і відпочинком.

Закон України “Про охорону праці” [3] визначає основні принципи, завдання, права і обов'язки суб'єктів відносин з охорони праці, а також організаційні та правові основи державного управління і контролю за дотриманням законодавства про охорону праці.

Згідно з цим законом, ІТ компанії повинні впроваджувати такі заходи з охорони праці:

1. Створювати на підприємстві службу охорони праці або призначати відповідальних осіб, які забезпечують розроблення, реалізацію та контроль за дотриманням заходів з охорони праці.

2. Забезпечувати безпечні і нешкідливі умови праці для працівників,

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		100

використовуючи сучасні засоби техніки безпеки, санітарно-гігієнічні умови, засоби клектинго та індивідуального захисту, оптимальні режими праці та відпочинку.

3. Проводити атестацію робочих місць на відповідність нормативно-правовим актам з охорони праці та аудит з охорони праці.

4. Проводити навчання та інструктаж з питань охорони праці, з надання першої медичної допомоги потерпілим від нещасних випадків і правил поведінки у разі виникнення аварії.

5. Забезпечувати лікувально-профілактичне обслуговування працюючих, санітарно-побутове обслуговування, пільги і компенсації для працівників, які працюють у важких і шкідливих умовах.

6. Нести відповідальність за порушення законодавства про охорону праці та зподіяння шкоди життю і здоров'ю працівників.

8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

Санітарно-гігієнічні вимоги щодо влаштування приміщень, в яких працівники працюють на персональних комп'ютерах, параметрів виробничого середовища, організації та обладнання робочих місць, режимів праці та відпочинку при роботі з відеотерміналами (ВДТ) і профілактичних медоглядів наведено у ДСанПіН 3.3.2.007-98 «Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин», затверджених постановою Головного державного санітарного лікаря України від 10 грудня 1998 р. № 7 (далі – ДСанПіН 3.3.2.007-98) [6].

Згідно ДСанПіН 3.3.2.0007-98 [6], шкідливі і небезпечні фактори при роботі з комп'ютером можуть бути такі:

1. Електромагнітне випромінювання – це випромінювання, яке створюється комп'ютером і його периферійними пристроями, такими як монітор, пртер, сканер тощо. Це випромінювання може викликати головний біль, запаморчення, розлади сну, зниження імунітету та інші негативні ефекти.

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		101

2. Електростатичне поле – це поле, яке утворюється внаслідок накопичення електричних зарядів на поверхні комп'ютера і його частин. Це поле може спричиняти сухість шкіри, свербіж, подразнення очей, алергічні реакції та інші проблеми.

3. Нервово-емоційне напруження – це напруження, яке виникає внаслідок тривалої концентрації уваги, високої вимогливості до результату роботи, нестабільності програмного забезпечення, конфліктних ситуацій тощо. Це напруження може призводити до стресу, депресії, роздратування, погіршення пам'яті та інших порушень.

4. Навантаження на органи зору – це навантаження, яке виникає внаслідок довго перебування перед монітором, низької якості зображення, недостатнього освітлення приміщення, поганої ергономіки робочого місця тощо. Це навантаження може спричиняти зниження гостроти зору, появу блимавок і кругів перед очима, синдром сухого ока та інші захворювання.

5. Дрібні стереостатичні рухи кінцівок – це рухи, які пов'язані з керуванням клавіатурою і мишею. Ці рухи можуть призводити до перевантаження і запалення суглобів і сухожилів рук і пальців, а також до тендовагінітів і синдрому зап'ясткового каналу.

Таким чином, робота з комп'ютером не така безпечна, як може здатися на перший погляд. Тому дуже важливо дотримуватися правил охорони праці і гігієни при роботі з комп'ютером.

8.3 Розробка заходів з умов поліпшення охорони праці

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

- розмірі приміщення, у розрахунку на одному працюючого, відповідають нормативам;
- мікроклімат відповідає нормативному значенню;

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		102

– 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).

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

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

8.4 Пожежна безпека

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

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		103

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

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

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

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

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

Також встановлюється порядок роботи та технічного обслуговування:

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

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

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		104

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

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

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

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

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

Робота обслуговуючого персоналу полягає в інсталяції необхідного програмного забезпечення й наступному його використанні в діалоговому режимі роботи з ЕОМ. Іноді може виникати необхідність написання допоміжних програм для поліпшення роботи вузла або для зниження витрат. З погляду забезпечення умов праці й вимог техніки безпеки для роботи програміста необхідно наступне: достатнє висвітлення екрана дисплея й робочого місця; повна технічна справність

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		105

устаткування, його електробезпеку; достатня пожежна безпека приміщення; оптимальний мікроклімат, що сприяє продуктивній роботі; відповідність робочого місця вимогам ергономіки. До небезпечних і шкідливих факторів, дії яких піддається програміст, можна віднести: можливість поразки електричним струмом, при електроні справності встаткування, порушенні заземлення або техніки безпеки; робота в мікрокліматі з неприпустимими параметрами; робота при недостатній освітленості екрана дисплея й робочого місця.

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

Персональні ЕОМ можна віднести до першого класу електротехнічних виробів по способі захисту людини від поразки електричним струмом, оскільки їхні корпуси зроблені з ізолюючої пластмаси й кожен пристрій має заземлення. Відповідно правилам пристрою електроустановок ЕОМ можна віднести до електроустановок з робочою напругою до 1000 В.

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

Необхідно передбачити наявність у межах досяжності первинних засобів гасіння пожежі (вогнегасників) для локалізації вогню власними засобами до приїзду команди пожежної охорони. Повинен бути розроблений план екстреної евакуації персоналу при виникненні загоряння. Кількість евакуаційних виходів

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		106

повинне бути не менш двох. Допускається використання одного евакуаційного виходу, якщо відстань найбільш віддаленого робочого місця до цього виходу не перевищує 25 м.

8.5 Розрахунок занулення

Розрахунок занулення складається з трьох частин:

1. Розрахунок на відключаючу спроможність.
2. Визначення максимальної напруги на корпусі обладнання відносно землі при замиканні фази на корпус.

3. Розрахунок робочого і повторного заземлювачів

Початкові дані

1. Потужність електроприладів, які підлягають зануленню:

$$P = 11 \text{ кВт.}$$

2. Довжина магістрального кабеля: $L_M=35 \text{ м.}$

3. Довжина розгалудження (від розподільчого щита до електроприладів) :
 $l=25 \text{ м.}$

4. Матеріал провідників кабеля—алюміній.

5. Лінійна напруга $U=380 \text{ В.}$

6. Фазна напруга $U_f=220 \text{ В.}$

Розрахунок проводиться відповідно до схеми електромережі, яка зображена на рис. 8.1:

У результаті розрахунку отримали:

- номінальна силу струму апарата захисту 25 А.
- найменше допустиме по умовам спрацьовування захисту значення сили струму короткого замикання 45 А.
- площа перерізу магістрального кабеля (провідника) 6 мм^2 .
- максимальної напруги на корпусі обладнання відносно землі при замиканні фази на корпус 20,5 В.

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		107

рівнем вищої освіти для здобувачів вищої освіти спеціальностей 123 "Комп'ютерна інженерія" та 122 "Комп'ютерні науки" / М-во освіти і науки України, Центральнoукраїн. нац. техн. ун-т, каф. кібербезпеки та програм. забезпечення; [укл. О.В. Оришака, К.М. Марченко]. – Кропивницький: ЦНТУ, 2022. – 19 с. [Електронний ресурс]. – Режим доступу: <http://dspace.kntu.kr.ua/jspui/handle/123456789/12240> (дата звернення 19.09.23).

5. НПАОП 40.1-1.21-98. Правила безпечної експлуатації електроустановок споживачів – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/z0093-98#Text> (дата звернення 19.09.23).

6. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин: ДСанПІН 3.3.2-007-98. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/v0007282-98> (дата звернення 19.09.23).

КБПЗ – 2023

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		109

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи передачі мультимедійних даних за технологією IMS.

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 передачі мультимедійних даних за технологією IMS.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем передачі мультимедійних даних за технологією IMS.

– Досліджена система передачі мультимедійних даних за технологією IMS.

– На основі отриманих результатів досліджень створена програмна реалізація системи передачі мультимедійних даних за технологією IMS.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання передачі мультимедійних даних за технологією IMS.

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

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

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		110

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

Програма реалізована на мові високого рівня Delphi 10.4.1. Дана мова програмування дозволяє найбільш ефективно обробляти дані. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

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

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм Serpent.

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 4471 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 0,86 роки.

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		111

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Рябошапко О.О. Дослідження та програмна реалізація системи передачі мультимедійних даних за технологією IMS // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2023.

2. The 3G IP Multimedia Subsystem (IMS): Merging the Internet and the Cellular Worlds // Gonzalo Camarillo, Miguel-Angel García-Martín (John Wiley & Sons, 2006, ISBN 0-470-01818-6).

3. The IMS: IP Multimedia Concepts and Services // Miikka Poikselka, Aki Niemi, Hisham Khartabil, Georg Mayer (John Wiley & Sons, 2006, ISBN 0-470-01906-9).

4. ITU-T Recommendation G.723.1. Dual Rate speech coder for multimedia communication transmitting at 5.3 and 6.3 kbit / sec. – 1996.

5. ITU-T Recommendation G.729. Speech codec for multimedia telecommunications transmitting at 8 / 13 kbit / s. – 1996.

6. ITU-T Recommendation H.225.0. Call signaling protocols and media stream packetization for packet-based multimedia communication systems. -Geneva, 1998.

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

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

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

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

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

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		112

12. ITU-T Recommendation H.323. Packet based multimedia communication systems. – Geneva, 1998.
13. ITU-T Recommendation H.324. Terminal for Low Bit Rate Multimedia Communications. -1996.
14. ITU-T Recommendation Q.931. ISDN User-Network Interface Layer 3 Specification for Basic Call Control. – 1993.
15. RFC 2205. Resource Reservation Protocol (RSVP). Ver.1. Functional Specification. – September 1997.
16. RFC 2327. Session Description Protocol. M. Handley, V. Jacobson. April, 1998.
17. RFC 2543. SIP: Session Initiation Protocol. M. Handley, H. Schulzrinne, E. Schooler, J. Rosenberg. March 1999.
18. RFC 2705. Media Gateway Control Protocol (MGCP) Version 1.0. M. Arango, A. Dugan, I. Elliott, C. Huitema, S. Pickett. October 1999.
19. RFC 2865. Remote Authentication Dial In User Service (RADIUS). C.Rigney, S. Willens, A. Rubens, W. Simpson. June 2000.
20. RFC 2885. Megaco Protocol 0.8. A.Rayhan, B. Rosen, J. Segers. August 2000.
21. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». Lecture Notes on Data Engineering and Communications Technologies, 2023, 178, pp. 208–223.
22. Smirnov, O., Neskoriyeva, T., Fedorov, E., Rudakov, K., Neskoriyeva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» CEUR Workshop Proceedings, Volume 3187, 2022,
23. Smirnov O., Smirnova T., Anas M. Al-Oraiqat, Drieiev O., Polishchuk L., Sheroz Khan, Yassin M. Y. Hasan, Aladdein M. Amro, Hazim S. AlRawashdeh «Method for Determining Treated Metal Surface Quality Using Computer Vision Technology». Sensors (Basel, Switzerland) Volume 22, Issue 16, 6223, 2022.

					БКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		113

24. Smirnov O., Kuznetsov A., Kryvinska N., Kiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». SN Computer Science, Vol 2, 337, 2021. <https://doi.org/10.1007/s42979-021-00739-w>

25. Smirnov O., Kuznetsov A., Zhora V., Onikiychuk A., Pieshkova O. «Hiding Messages in Audio Files Using Direct Spread Spectrum». 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2021, Cracow, Poland, 22-25 September 2021. P. 414-418.

26. Smirnov O., Kuznetsov A., Lokotkova I., Kuznetsova T., Florov S., Lebid O. «Using Orthogonal Signals to Hide Information in Images». 4 IEEE International Conference on Advanced Information and Communication Technologies (AICT) – 2021, Lviv, Ukraine, September 21-25, 2021. P. 255-260.

27. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256.

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

29. Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». Journal of theoretical and applied information technology Vol.98. No 21, 2020, P. 3334-3346.

30. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». CEUR Workshop Proceedings Volume 2654, 2020, Pages 1-14.

31. Smirnov O., Kuznetsov A., Onikiychuk A., Makushenko T., Anisimova O., Arischenko A. «Adaptive pseudo-random sequence generation for spread spectrum image steganography». 2020 IEEE 11th International Conference on Dependable

					БКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		114

Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 161-165.

32. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 172-177.

33. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 48. Springer, Cham. 2021. pp 557-587.

34. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». CEUR Workshop Proceedings Volume 2616, 2020, Pages 366-379.

35. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», CEUR Workshop Proceedings Volume 2608, 2020, Pages 633-645.

36. Smirnov, O., Kuznetsov, A., Gorbacheva, L., Babenko, V., «Hiding data in images using a pseudo-random sequence», CEUR Workshop Proceedings Volume 2608, 2020, Pages 646-660.

37. Zhurakovskiy, B., Tsopa, N., Batrak, Y., Odarchenko, R., Smirnova, T «Comparative analysis of modern formats of lossy audio compression». Workshop Proceedings, 2020, 2654, стр. 315-327.

38. Smirnov O. Kuznetsov A., Zaichenko Yu., Pastukhov M., Oleshko O., Kuznetsova K., «Formation of Discrete Signals with Special Correlation Properties». International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2019; Odessa; Ukraine; 9-13 September 2019. P.22-28.

					БКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		115

39. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». International Journal of Computing; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.

40. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), Kyiv, Ukraine, 8 – 11 October 2019 . P.517-522.

41. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». CEUR Workshop Proceedings, Vol 2588, P. 90-106, 2019.

42. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Averchev, A., Pastukhov, M., Kuznetsova, K., «Formation of Pseudorandom Sequences with Special Correlation Properties», 2019 3rd International Conference on Advanced Information and Communications Technologies, AICT -2019/ Lviv, Ukraine, 2-6 July, 2019, P. 395-399.

43. Smirnov, O., Kuznetsov, A., Kavun, S., Babenko, B., Nakisko, O., Kuznetsova, K., «Malware Correlation Monitoring in Computer Networks of Promising Smart Grids», 2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS), Kyiv, Ukraine April 17-19, 2019 P. 347-352.

44. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special Correlation Properties», CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 618-629.

45. Smirnov A.A., Kuznetsov A.A., Danilenko D.A., Berezovsky A., «The statistical analysis of a network traffic for the intrusion detection and prevention systems», Telecommunications and Radio Engineering. – Volume 74, Issue 1. – Begel House Inc. – 2015. – P. 61-78.

46. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New Technique for Hiding Data in Cover Images Using Adaptively Generated Pseudorandom Sequences». CEUR Workshop Proceedings Volume 2732, 2020, Pages 214-227.

47. Т.В. Смірнова, О.М. Дреєв, О.А. Смірнов «Хмарна інформаційна система оцінювання шорсткості з використанням дискретного частотного аналізу макрофотографій». IV міжнародна науково-практична конференція «Інформаційна безпека та комп'ютерні технології», м. Кропивницький. 15-16 квітня 2021р. – Кропивницький: ЦНТУ. – 2021. – С. 30.

48. О.А. Смірнов, П.С. Усік, «Дослідження перспектив використання технологічних рішень в мережах 5G» у Кібербезпека та інформаційні технології: монографія. – Х. : ТОВ «ДІСА ПЛЮС», 2020.С. 122-135.

49. О.А.Смірнов, Т.В.Смірнова, Л.І. Поліщук, К.О. Буравченко, А.О.Макевнін, «Дослідження хмарних технологій як сервісів», Кібербезпека: освіта, наука, техніка. № 3(7). С. 43-62. 2020.

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

					ВКРМ-122.23.0080.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		117

Додаток А
(обов'язковий)

Технічне завдання

Зміст

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

					ВКРМ-122.23.0080.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Рябошапко О.О.				<i>Дослідження та програмна реалізація системи передачі мультимедійних даних за технологією IMS</i>	Літ.	Аркуш	Аркушів
Перевірів	Дресва Г.М.					М	1	6
Н. Контр.	Коваленко А.С.				ЦНТУ КН-22МЗ			
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи передачі мультимедійних даних за технологією IMS.

2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 37-13 від 04.08.2023 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи передачі мультимедійних даних за технологією IMS.

4 Джерела розробки

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

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

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

					ВКРМ-122.23.0080.00.00.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 передачі мультимедійних даних за технологією IMS;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

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

5.4 Вимоги до архітектури

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

5.5 Вимоги до надійності

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

					ВКРМ-122.23.0080.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

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

5.7 Вимоги до складу та параметрів технічних засобів

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

5.8 Вимоги до інформаційної і програмної сумісності

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

5.8.1 Обладнання

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

5.8.2 Мова програмування

Середовище Delphi 10.4.1.

					ВКРМ-122.23.0080.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

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

7 Економічні вимоги

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

7.2 Виконати розрахунок витрат показників економічного ефекту з урахуванням цін на 3 вересня 2023 року.

8 Вимоги щодо охорони праці

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

					ВКРМ-122.23.0080.00.00.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 – 117 аркушів.

10 Етапи розробки

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

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

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

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Розрахунок з техніко-економічного обґрунтування.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

11.1 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на попередній захист 10.12.2023 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 18.12.2023 р.

					ВКРМ-122.23.0080.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

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

_____ Дреєва Г.М.

*Дослідження та програмна реалізація
системи передачі мультимедійних даних за технологією IMS*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 34

Літера: РП

Кропивницький – 2023 року

Програма організації зв'язку

```

program IMS;

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 := 'IMS';
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.
```

КБПЗ_2023

Підпрограма завантаження вікна вікна Завантажень

```
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.
```

КБПЗ_2023

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;
    MainMenu: 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;
    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;
    Imagel: 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#$T%&(*qwrda@@@#45';
    {$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);
      end;
    end;
  end;
end;

```

```

        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 знайдено!',mtInformation,[mbOK],0);
    end else
    begin
        ListBox2.Items.LoadFromFile('MainLegend.dat');
    end;
    if FileExists('MainIP.dat')=false then
    begin
        MessageDlg('Файл MainIP.dat не знайдено!',mtInformation,[mbOK],0);
    end else
    begin
        ListBox1.Items.LoadFromFile('MainIP.dat');
    end;
    if FileExists('MainHISTORY.dat')=false then MessageDlg('Файл MainHISTORY.dat
не знайдено!',mtInformation,[mbOK],0)
    else ListBox3.Items.LoadFromFile('MainHISTORY.dat');
    if FileExists('mainHelp.dat')=false then MessageDlg('Файл mainHelp.dat не
знайдено!',mtInformation,[mbOK],0)
    else Memol.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
знайдено!',mtInformation,[mbOK],0)
        else ListBox2.Items.SaveToFile('MainLegend.dat');
        if FileExists('MainIP.dat')=false then MessageDlg('Файл MainIP.dat не
знайдено!',mtInformation,[mbOK],0)
        else ListBox1.Items.SaveToFile('MainIP.dat');
        if FileExists('MainHISTORY.dat')=false then MessageDlg('Файл MainHISTORY.dat
не знайдено!',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.
```

К6П3_2023

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].Data.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.
```

К6ПЗ_2023

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.
```

КБПЗ_2023

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;
    Animatel: 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.
```

Підпрограма парольного захисту

```

unit Unit4;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Buttons, Mask, ExtCtrls;
type
  TForm4 = class(TForm)
    Panell: 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#$T%&(*qwrda@@@#45';
  DAT:string='VUU';
{$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.

```

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 (Pntr:Pointer);
begin
  Items.Add (Pntr);
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;
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;

{-----}
{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;

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;

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 (FDestinations)) 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;
            end;
          end;
        end;
      end;
    end;
  end;
end;

```

```

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;

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
                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
                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;
            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;

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

        A := 0;
        while (not VolSet OR not MuteSet) AND (A < Cntrl.Count) do
        begin
            Cntrl := Cntrl[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;
    pmcMixerDataUnsignedMute : 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(pmcMixerDataUnsignedMute, 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;
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;
          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(pmcMixerDataUnsignedMute);
FreeMem(pmcMixerControlMute);
end;
end;

function TAudioMixer.SetMute (ADestination, AConnection: Integer; Mute:
Boolean): Boolean;
var
    MD : TMixerDestination;
    MC : TMixerConnection;
    mlcMixerLineControlsMute : TMIXERLINECONTROLS;
    mcdMixerDataMute : TMIXERCONTROLDETAILS;
    pmcMixerControlMute : PMIXERCONTROL;
    pmcMixerDataUnsignedMute : 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;
        end;

    GetMem(pmcMixerControlMute, SizeOf(TMIXERCONTROL));
    GetMem(pmcMixerDataUnsignedMute, 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 (pmcMixerDataUnsignedMute);
FreeMem (pmcMixerControlMute);
end;
end;

function TAudioMixer.GetPeak (ADestination, AConnection:Integer; var LeftPeak,
RightPeak:Integer): Boolean;
var
  MD : TMixerDestination;
  MC : TMixerConnection;
  mcdMixerDataPeak : TMIXERCONTROLDETAILS;
  pmcMixerControlPeak : PMIXERCONTROL;
  { pmcMixerDataSignedPeak : PMIXERCONTROLDETAILSSIGNED;}
  mlMixerLine : TMixerLine;
  A:Integer;
  Cntrl: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;
          Cntrl := MD.Controls;
        end
      else
        begin
          MC := MD.Connections[AConnection];
          if MC <> nil then
            begin
              mlMixerLine := MC.Data;
              Cntrl := 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.

```