

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

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

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи надання
доступу до мережі Internet сервіс провайдера”

Виконав здобувач вищої освіти
II курсу, групи КІ-22М-1
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Штогрін Б.В.
« ____ » _____ 2023 р.

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

Центральноукраїнський національний технічний університет
Факультет *Механіко-технологічний*
Кафедра *Кібербезпеки та програмного забезпечення*
Рівень вищої освіти *магістр*
Галузь знань *12* "Інформаційні технології"
Спеціальність *123* "Комп'ютерна інженерія"
Освітньо-професійна (освітньо-наукова) програма "Комп'ютерна інженерія"

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

Штогрину Богдану Васильовичу

(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження та програмна реалізація системи надання доступу до мережі Internet сервіс провайдера*

2. Керівник роботи *Кислун Олег Андрійович, канд. техн. наук, доцент*

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

затверджені наказом вищого навчального закладу № 34-13 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 надання доступу до мережі Internet сервіс провайдера*

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

- | | |
|--|---|
| <i>1. Призначення та область використання.</i> | <i>6. Наукова новизна.</i> |
| <i>2. Перегляд аналогічних існуючих систем.</i> | <i>7. Економічна ефективність розробленої програми.</i> |
| <i>3. Опис і обґрунтування проектних рішень.</i> | <i>8. Заходи з охорони праці та техніки безпеки.</i> |
| <i>4. Етапи програмування системи.</i> | <i>9. Висновки.</i> |
| <i>5. Впровадження системи в промислову експлуатацію</i> | |

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

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

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Савеленко Г.В.	05.10.2023	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

Штогрін Б.В. Дослідження та програмна реалізація системи надання доступу до мережі Internet сервіс провайдера. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2023.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи надання доступу до мережі Internet сервіс провайдера.

Метою розробки є дослідження та програмна реалізація системи надання доступу до мережі Internet сервіс провайдера.

Об'єктом дослідження є процес надання доступу до мережі Internet сервіс провайдера.

Предметом дослідження є методи надання доступу до мережі Internet сервіс провайдера.

Методи дослідження базуються на методах теорії комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи надання доступу до мережі Internet сервіс провайдера.

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 PHP, Perl.

Ключові слова: комп'ютерна інженерія, Internet сервіс провайдера

ABSTRACT

Shtohrin B.V. Research and software implementation of the system for providing access to the Internet service provider's network. 123 Computer engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2023.

In this graduation thesis for the second (master's) level of higher education, software is developed, which is intended for the system of providing access to the Internet service provider's network.

The purpose of the development is the research and software implementation of the system for providing access to the Internet service provider's network.

The object of the study is the process of providing access to the Internet service provider's network.

The subject of the study is methods of providing access to the Internet service provider's network.

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

The result of the work is the software implementation of the system for providing access to the Internet service provider's network.

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 PHP, Perl environment.

Keywords: computer engineering, Internet service provider

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	6
1.1 Призначення системи.....	6
1.2 Область застосування.....	8
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	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.....	19
2.3 Розгорнута постановка завдання	23
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	25
3.1 Опис функціонування системи	25
3.2 Розробка структурної схеми.....	35
3.3 Розробка функціональної схеми	42
3.4 Розробка діаграми процесів.....	45
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	47
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	47
4.2 Захист розробленого програмного забезпечення.....	59
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	66
6 НАУКОВА НОВИЗНА	68

					ВКРМ-123.23.0024.00.00.ПЗ			
Вим	Арк.	№ докум.	Підп.	Дата	Дослідження та програмна реалізація системи надання доступу до мережі Internet сервіс провайдера	Літ.	Аркуш	Аркушів
Розроб.	Штогрін Б.В.					М	1	104
Перев.	Кислун О.А.					ЦНТУ КІ-22М-1		
Н.контр.	Коваленко А.С.							
Затв.	Смірнов О.А.							

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	69
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	69
7.2 Розрахунок трудомісткості розробки програмної продукції.....	71
7.3 Визначення чисельності виконавців і планового фонду зарплати.....	73
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника.....	77
7.5 Визначення собівартості розробки та ціни програмної продукції.....	82
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	85
7.7 Визначення експлуатаційних витрат.....	86
7.8 Визначення економічної ефективності програмної продукції.....	87
7.9 Висновок.....	89
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	90
8.1 Вступ.....	90
8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером.....	91
8.3 Пропозиції щодо підвищення працездатності ІТ-фахівця.....	92
8.4 Розрахункова частина	93
8.5 Висновки до розділу.....	95
9 ОСНОВНІ ВИСНОВКИ.....	96
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	98

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

HTML (HyperText Markup Language) – мова розмітки гіпертекстових документів;

СУБД – системи управління базами даних;

TCP/IP (Transport Control Protocol/Internet Protocol) – протокол роботи мережі Інтернет;

ГМ – глобальні мережі;

ЛВМ – локальна віртуальна мережа;

ОД – остаточне обладнання даних;

ОС – операційна система;

ПК – персональний комп'ютер;

ОЗП – основний запам'ятовуючий пристрій;

URL – universal resource locator – локатор ресурсів інтернет;

PPTP – Point-to-Point-Tunneling Protocol – протокол створення захищеного каналу при доступі віддалених користувачів через публічні мережі;

ВСЗП – віртуальна система захисту інформаційного потоку;

ВСПП – віртуальні системи поділу інформаційних потоків;

ВДТ – відеодисплейний термінал;

ЦО – цивільна оборона;

ПЕОМ – персональна електронно обчислювальна машина;

PHP – створює HTML-сторінки з використанням спеціальних тегів, розпізнаваних аналізатором PHP;

WWW – world wide web – всесвітня мережа інтернет;

FTP – File Transport Protocol – протокол передачі файлів.

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

Актуальність теми. Інтернет-провайдер (інтернет-провайдер) – це компанія, яка надає окремим особам і організаціям доступ до Інтернету та інших пов'язаних послуг. Інтернет-провайдер має обладнання та доступ до телекомунікаційної лінії, необхідні для того, щоб мати точку присутності в Інтернеті для географічної області, що обслуговується. Інтернет-провайдери надають клієнтам доступ до Інтернету, а також надають додаткові послуги, такі як електронна пошта, реєстрація домену та веб-хостинг. Інтернет-провайдери також можуть надавати різні типи підключення до Інтернету, наприклад кабельне та оптоволоконне. Підключення також може бути високошвидкісним широкосмуговим або неширокосмуговим. Федеральна комісія зі зв'язку (FCC) стверджує, що для того, щоб вважатися високошвидкісним, з'єднання має мати швидкість завантаження не менше 25 мегабіт на секунду (Мбіт/с) і швидкість завантаження не менше 3 Мбіт/с. Інтернет-провайдера також іноді називають *постачальником доступу до Інтернету*. ISP також іноді використовується як аббревіатура для *незалежного постачальника послуг*, щоб відрізнити постачальника послуг, який є окремою компанією від телефонної компанії.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи надання доступу до мережі Internet сервіс провайдера.

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

- Огляд існуючих систем надання доступу до мережі Internet сервіс провайдера.
- Дослідження системи надання доступу до мережі Internet сервіс провайдера.
- Програмна реалізація системи надання доступу до мережі Internet сервіс провайдера.

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

Об'єктом дослідження є процес надання доступу до мережі Internet сервіс провайдера.

Предметом дослідження є методи надання доступу до мережі Internet сервіс провайдера.

Методи дослідження базуються на методах теорії комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.

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

– Удосконалено метод надання доступу до мережі Internet сервіс провайдера.

– Розроблено вітчизняний продукт надання доступу до мережі Internet сервіс провайдера, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі надання доступу до мережі Internet сервіс провайдера.

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

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

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи надання доступу до мережі Internet сервіс провайдера, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Система призначена для надання доступу до мережі Інтернет сервіс провайдера. Для виходу в інтернет при використанні з'єднання ADSL необхідні модем і телефонна лінія. Додатковий кабель при цьому не прокладається. З устаткування потрібний модем, що підключається до телефонної лінії й до комп'ютера. Багато провайдерів надають модеми в оренду, якщо ви підписуєте договір на досить дорогий пакет, то оренда буде фактично безкоштовною. Підключення полягає в тому, що клієнтові надається паралельна телефонної лінія, потім певним чином налаштовується модем. Досить стабільне з'єднання, хоча швидкості все-таки обмежені, та й імовірність того, що сигнал буде перериватися у випадку перевантаження лінії або при несприятливих погодних умовах досить висока.

Підключення виділеної лінії здійснюється за допомогою окремого спеціального кабелю. При цьому телефонна лінія вже не задіється, у модемі немає необхідності. Таке з'єднання відрізняється високою швидкістю й можливістю надавати безлімітні пакети по самих доступних цінах. Якщо ви не знаєте, із чого почати і як підключити домашній інтернет, то насамперед з'ясуйте, є чи у вашій місцевості провайдер, що надає послуги кабельного інтернету. Підключення відбувається в такий спосіб: до будинку прокладається оптоволоконний кабель, до квартири – кручена пари. Налаштування комп'ютера займає не більше напівгодини.

Багато хто вважають, що якщо будинок входить у зону покриття Wi-Fi, то залишається тільки придбати картку провайдеру, що надає послуги бездротового інтернету. Але не все так просто. Зв'язок може бути нестабільним, а сигнал –

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

слабким. Можна, звичайно купити Wi-Fi-адаптер для посилення сигналу, але при нестабільному покритті такий хід навряд чи що-небудь змінить.

Супутниковий інтернет дуже дорогий й громіздкий, тому для жителів великих міст неприйнятний, з огляду на чималий вибір інших варіантів. А от у пригороді, коттеджних і дачних селищах супутник серйозно виручить. Швидкість порядку 4 Мб/с, можна приймати також супутникові телеканали. Налаштування складна, котру проводить майстер, як правило, протягом декількох днів. Інформація приймається із супутника, а передача здійснюється по іншому з'єднанню (GPRS, DSL, Dial-Up) наземним провайдером. Для з'єднання необхідна DVB-карта аналог модему), що обробляє дані, одержувані із супутника.

Інтернет від мобільних операторів зручно мати на крайній випадок, але як постійний доступ він не підійде: незважаючи на те що завжди під рукою, однак це дорого й нестабільно. Найбільше поширення одержали два стандарти, які підтримують передачу даних: CDMA і GSM. Для підключення необхідні стільниковий телефон або спеціальна форма оплати.

Безлімітний інтернет – найбільш затребувана сьогодні послуга серед тих, хто підключає до Мережі свій домашній комп'ютер. Не дивно: доступ в інтернет надається без обмежень, тобто обсяг завантажуючої інформації й час, що було проведено в інтернеті, необмежений. Користувачеві необхідно внести фіксовану плату за певний період, після чого можна не виходити з інтернету весь оплачений період і качати скільки душі завгодно інформації.

Варіантів, як підключити безлімітний інтернет, декілька. По-перше, такий пакет пропонують всі провайдери, які надають кабельне оптоволоконне з'єднання. Єдине обмеження – швидкість, але вона досить висока (можна без проблем дивитися відео онлайн) навіть при покупці початкових за ціною й доступних уже, напевно, для кожного користувача, пакетів.

По-друге, відносно недавно послугу безлімітного доступу стали пропонувати своїм клієнтам оператори мобільних телефонів. Це 2G/2,5G, 3G і 4G інтернет. Питання, як підключити безлімітний інтернет, у цій ситуації

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

вирішується просто: необхідно придбати в мобільного оператора, що є провайдером, спеціальний модем, вибрати й оплатити пакет послуг.

Сьогодні можна запросто зайти в інтернет з будь-якого телефону за допомогою мережі 2G/2,5G через операторів, що забезпечують мобільний зв'язок. По суті 2G/2,5G являє собою добре відомий всім користувачам GPRS (пакетний радіозв'язок загального користування). Цей тип зв'язку функціонує скрізь, де є покриття оператора мобільної мережі. Правда, у цієї послуги є істотний недолік: із всіх видів бездротового зв'язку швидкість передачі даних найнижча: від 171,2 до 474 кбіт/с.

А от за допомогою послуги 3G або 4G інтернет можна без проводів і кабелів підключитися до комп'ютера, планшету, ноутбуку практично в будь-якій точці. Технологія мобільного зв'язку 3-го покоління 3G надає швидкість, близько 3,6 Мбіт/с. Це достатня швидкість для того, щоб переглядати сайти, пошту, спілкуватися в соцмережах, дивитися відео онлайн і завантажувати аудіо– і текстові файли. Ця технологія продовжує вдосконалюватися з кожним днем, у цей час покриття забезпечується дуже значне, а отже, популярність цієї послуги серед користувачів зростає.

Варто згадати про перспективний 4G (четверті покоління) мобільного зв'язку, що припускає найвищу на сьогоднішній день швидкість: 100 Мбіт/с рухливим абонентам і 1 Гбіт/с – стаціонарним, що дає можливість швидкої роботи в Інтернеті. Дана технологія поки ще тільки починає своє входження на ринок інтернет-послуг.

1.2 Область застосування

Областю застосування програмного забезпечення, яке розробляється у даному магістерському проекті є інтернет сервіс провайдери. Організація, що надає послуги, пов'язані з доступом до інтернету, називається інтернет-провайдером. Простіше говорячи, це постачальники інтернету.

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

Основні види доступу:

- широкополосний;
- що комутується;
- бездротовий.

Крім того, може бути виділений дисковий простір для зберігання й роботи сайтів, забезпечуватися підтримка поштового сервера або поштових скриньок. Провайдер також надає місце на своїй площадці для розміщення устаткування, пропонує оренду віртуальних і виділених серверів.

Інтернет-провайдер (постачальник інтернет-послуг) – організація, що надає доступ до інтернету й інші супутні послуги. З погляду права це оператор зв'язку, що має ліцензію на такі види послуг:

- надання каналів зв'язку;
- передача даних у мережі (за винятком голосової інформації) ;
- передача голосової інформації в мережі при передачі даних;
- телематичний зв'язок.

Крім цього, інтернет-провайдер надає такі послуги:

- доступ, що комутується, в Інтернет;
- широкополосний доступ;
- бездротовий доступ;
- надання дискового простору, що забезпечує можливість зберігати дані й підтримувати роботу сайтів (хостинг);
- підтримка віртуального поштового сервера або поштової скриньки (однієї або декількох);
- розміщення на площадці, що належить провайдеру, клієнтського устаткування;
- резервування даних;
- оренда серверів (віртуальних і виділених (VPS, VDS)).

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

Відповідно до тих послуг, які надаються, можна виділити такі групи інтернет-провайдерів:

- хостинг-провайдери;
- провайдери доступу;
- магістральні;
- канальні.

Якщо говорити про провайдери доступу, тобто первинні (магістральні), що мають у власності магістральні канали, і вторинні (міські), що орендують у первинних провайдерів канали зв'язку. Первинні продають трафік, як правило, тільки в дуже великих обсягах і послуги роблять не індивідуальним користувачам, а переважно іншим провайдерам, хоча бувають, звичайно, і виключення.

Провайдер, що надає послуги мобільної комерції (Mobile Commerce Service Provider) – компанія, що пропонує ПЗ й спеціальні системи, використовувані на мобільних платформах електронної комерції, щоб установлювати їх на мобільних пристроях (телефонах, смартфонах). Провайдер мобільної комерції також робить мобільний контент і займається його продажем, створює мобільні веб-сайти, надає консультаційні послуги, а також послуги з інтеграції мобільних платежів у ЗМІ й Інтернеті. Іншими словами, завдяки таким провайдерам мобільні пристрої застосовуються тепер як засоби безготівкового розрахунку, тому за допомогою мобільного телефону можна оплачувати найрізноманітніші товари й послуги.

Сферами діяльності провайдерів мобільної комерції є:

- білінг-системи;
- апаратно-програмні розробки;
- сервіси текстових повідомлень;
- мобільні платежі;
- онлайн-маркетинг;
- створення веб-сайтів і хостинг.

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

Існують також інформаційні провайдери – компанії або фізособи, які від імені інших осіб одержують, відправляють, зберігають документи в електронному виді або надають спектр послуг щодо таких документів.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи надання доступу до мережі Internet сервіс провайдера, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

КБГПЗ - 2023

					VKPM-123.23.0024.00.00.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

Інтернет Контроль Сервер

Інтернет Контроль Сервер – програма для обліку трафіку та моніторингу інтернет підключення

Універсальний Інтернет-шлюз з інструментами для захисту корпоративної мережі, обліку трафіку, управління доступом, розгортання поштового, проксі, файлового сервера, Web та jabber сервера, організації IP-телефонії.

Інтернет Контроль Сервер (ІКС) – програма для обліку трафіку та моніторингу Інтернет підключення. Може використовуватись не тільки як лічильник трафіку, але і як універсальний комплекс для управління будь-яким підключенням до Інтернету.

Інтернет Контроль Сервер потрібно увімкнути між вашою корпоративною мережею та провайдерами, щоб обліковувати трафік з інтернету, керувати доступом кожного користувача, встановити надійний захист мережі.

Інтернет Контроль Сервер дозволить:

- переглядати хто, і для чого використовує підключення до Інтернет;
- вести підрахунок трафіку за допомогою зручного лічильника трафіку;
- обмежити Інтернет трафік, заборонивши співробітникам його нецільове витрачання;
- зменшити розтрата робочого часу;
- уникнути витоків інформації через Інтернет;
- не допустити вторгнення у мережу;
- контролювати ділове листування;
- організувати безпечний зв'язок між віддаленими офісами;

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

– забезпечити віддалений доступ співробітників організації в мережу компанії.

Інтернет Контроль Сервер включає всі можливі служби та сервіси сучасного підприємства для роботи в локальній мережі і Інтернет.

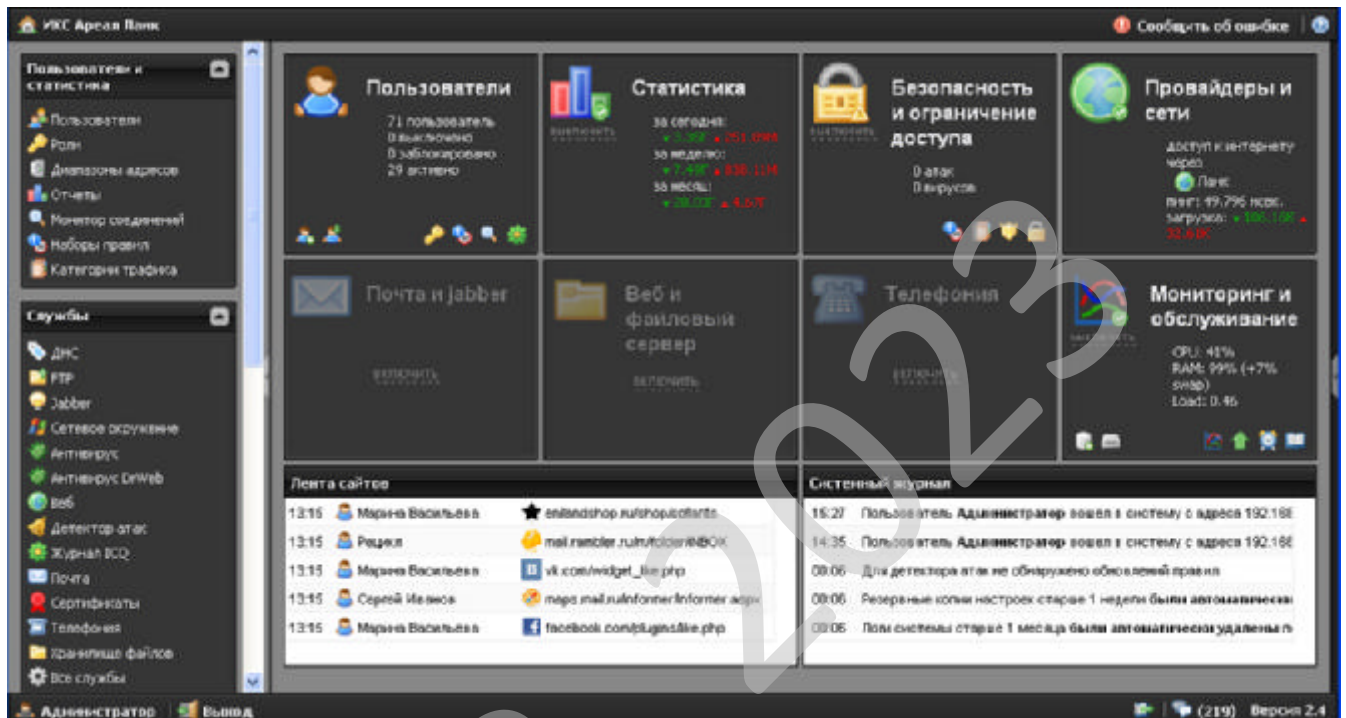


Рисунок 2.1 – Интерфейс користувача ПЗ «Інтернет Контроль Сервер»

VMExtreme

VMExtreme – це нова назва добре відомою багатьом програми Bandwidth Monitor. Раніше програма розповсюджувалася безкоштовно, тепер же вона має три версії, і безкоштовною є тільки базова. У цій версії не передбачено ніяких можливостей, окрім, власне, моніторингу трафіку, тому навряд чи можна вважати її конкурентом інших програм.

Версія VMExtreme Home пропонує трохи більше опцій для стеження за трафіком. Так, працюючи з нею, можна спостерігати за графіком швидкості передачі і прийому даних, вести лог-файли за різні періоди (день, тиждень,

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

місяць), захищати програму паролем від доступу інших користувачів до її налаштувань, створювати окремі профілі для різних користувачів.

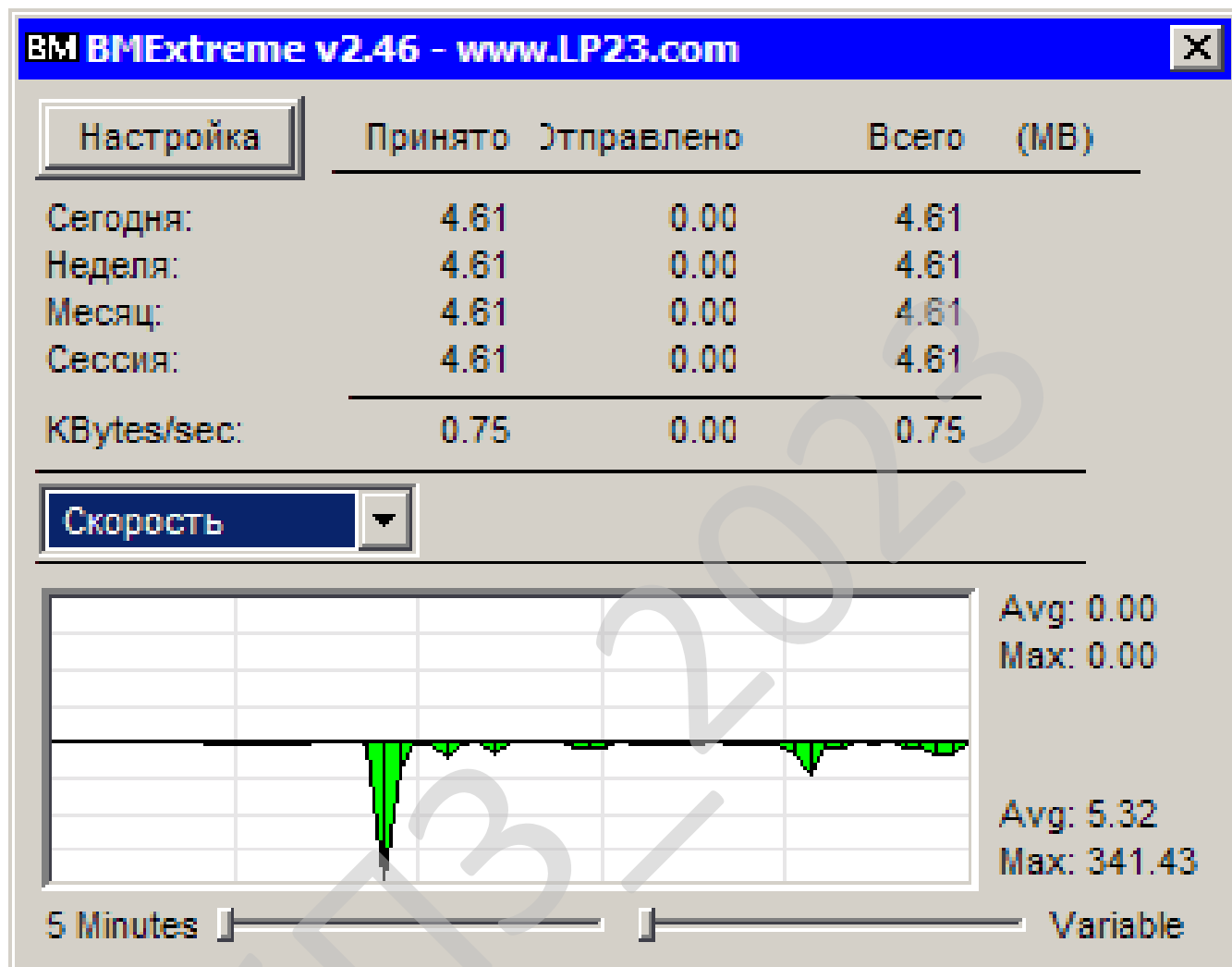


Рисунок 2.2 – Інтерфейс користувача VMExtreme

За умовчанням VMExtreme стежить як за Інтернет-трафіком, так і за трафіком в локальній мережі, проте моніторинг в LAN при бажанні можна відключити. За допомогою VMExtreme можна встановлювати обмеження на кількість отриманих або переданих даних, а також на загальний об'єм інформації, якою ви обмінюєтеся з Інтернетом.

Залежно від вашого тарифного плану, може знадобитися установка максимального об'єму даних, отриманих/переданих за останню добу, тиждень

або місяць. Є можливість і стежити за трафіком тільки у вказаний годинник. Як тільки програма зафіксує перевищення ліміту, вона може відключити з'єднання.

Версія VMExtreme Pro має ще гнучкіші настройки для моніторингу локального і Інтернет-трафіку. Наприклад, вона може вести статистику окремо по IP-адресах, що дуже зручно, якщо за прийом інформації в межах країни ви платите менше, ніж за світовий трафік.

ETI Software

Зростає потреба в інструментах керування для постачальників послуг бездротового Інтернету (WISP), оскільки клієнти очікують більше пропозицій і кращого обслуговування. Хороші інструменти керування допомагають постачальникам телекомунікацій у цьому обслуговуванні клієнтів, а також у розгортанні, обслуговуванні та розширенні їхніх пропозицій, включаючи з'єднання Wi-Fi та Ethernet-LAN для передачі відео, голосу, даних та Інтернету речей.

Хороша система управління допомагає організувати все так, щоб усі рівні організації могли працювати на одній сторінці, пропонуючи повний поточний огляд своїх пристроїв і послуг у будь-який момент часу.

Найкращі інструменти керування для постачальників послуг бездротового Інтернету побудовані на платформі, яка легко інтегрується з існуючими системами, усуваючи потребу в тривалому навчанні та налаштуванні. Завдяки платформі ETI телекомунікаційні компанії можуть використовувати програмне забезпечення, коли вони інтегрують нові послуги, будують свою мережу, масштабують її відповідно до потреб і змінюють її відповідно до нових правил і норм. Постачальникам послуг бездротового Інтернету більше не потрібно хвилюватися про те, що щось упустили або не помітять проблеми, оскільки вони бачать усе на зручній для читання геопросторовій інформаційній панелі, яка виділяє зони обслуговування, вказує на проблемні області та надає інформацію для планування та керування клієнтами.

Наші інструменти керування для постачальників послуг бездротового

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

Інтернету пропонують інтелектуальні, прості та функціональні дані в режимі реального часу, щоб допомогти оптимізувати взаємодію з клієнтами, спростити розгортання вантажівок і обмежити проблеми. Лише кількома клацаннями миші кожен, від агентів із обслуговування клієнтів до відділів продажів, може отримати інформацію про облікові записи передплатників, статуси робочих нарядів, доступ до прав, проблемні області та навіть життєвий цикл історії клієнтів.

Уніфіковане управління пристроями ЕТІ пропонує інструменти керування для постачальників послуг бездротового Інтернету для моніторингу та керування незалежними від постачальника пристроями вдома клієнта без необхідності фізично перебувати в цьому місці. Інформація автоматично збирається та повідомляється в одному місці, щоб представники служби підтримки клієнтів могли переглянути файл і знайти рішення, не вимагаючи від клієнта будь-яких дій. Тоді CSR мають можливість відстежувати підключення, пропускну здатність і результати в режимі реального часу. Після дзвінка клієнта він зможе переконатися, що проблему вирішено, і чи вплине ця проблема на інших клієнтів.

Інструменти керування ЕТІ для постачальників послуг бездротового Інтернету та програмне забезпечення працюють на всіх пристроях і в більшості випадків бездоганно інтегруються в існуючу систему моніторингу телекомунікаційної компанії. Вони скорочують роботу, яку повинні виконувати команди, яка не приносить цінності компанії. Це також допомагає організувати роботу та рух вантажівок. Більшість компаній готові вийти з коробки (або з полиці), немає потреби створювати або налаштовувати щось на замовлення.

З виходом нових стандартів більшість компаній зможуть одразу підключати свої пристрої. Якщо є потреба в додаткових інтеграціях, ми також можемо надати це.

Для постачальників послуг бездротового Інтернету головне – бути бездротовим. Головною перевагою програмного забезпечення для керування ЕТІ є те, що воно дає компаніям можливість віддалено підключатися до пристроїв,

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

Працює незалежно від виробника пристроїв.

Ця інформаційна панель може:

– Відстежуйте якість сигналів пристрою, щоб усунути проблеми під час встановлення.

– Відстежуйте, коли нові будівлі чи споруди викликають проблеми.

– Сигнал, коли дерева вирости на шляху сигналу.

– Список пристроїв і рейтинги підключення.

– Відстежує історичні дані, щоб надати розуміння.

– Надайте інформацію про можливості розширення.

– Допоможіть визначити нові бізнес-цілі.

– Відстежуйте витрати.

Крім того, Unified Device Management може надсилати оновлення одним натисканням кнопки, зменшуючи головний біль для CSR і технічних спеціалістів.

Інструмент масштабований і гнучкий, працює з різними пристроями та сервісами.

Платформа керування пристроями ETI Software, Unified Device Management, може допомогти компаніям будь-якого розміру та сфери діяльності.

Уніфіковане керування пристроями дає телекомунікаційним компаніям можливість координувати операції між відомою інформацією SAS і реальною статистикою радіочастот. Потім він відстежує SINR шляхом перехресного посилення на радіостатистику за час з EARFCN (абсолютний номер радіочастотного каналу E-UTRA). Незалежно від того, як змінюється чи змінюється SAS, уніфіковане керування пристроями може допомогти відстежувати пристрої та їхнє використання та виявляти проблеми до їх виникнення.

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Для написання програмного забезпечення була обрана мова програмування PHP.

PHP – це мова серверних скриптів (server scripting language), що вбудовується в HTML, який інтерпретується і виконується на сервері. PHP працює як частина Web-сервера, і цим самим схожий на ASP від Microsoft або Coldfusion від Allaire. Синтаксис PHP дуже схожий на синтаксис таких мов програмування C або Perl. Люди, що мають деякий досвід програмування, дуже швидко зможуть почати писати програми на PHP. У цій мові немає строгої типізації даних і немає необхідності в діях з виділення/звільнення пам'яті. Програми, написані на мові програмування PHP, читаються достатньо легко. На відміну від Perl-программ PHP-коду властива легка читабельність та зрозумілість.

PHP є препроцесором HTML. До того, як сервер відправить файл браузеру, його проглядає препроцесор-інтерпретатор. Для того, щоб це відбувалося, файли, які піддаються обробці препроцесором, повинні мати визначене розширення (звичайно це .phtml або .php3, але ці значення можна поміняти) і містити (хоча це не обов'язкова вимога) код для препроцесора. Перед відправкою сторінки PHP-код програється на сервері і браузеру видається результат у вигляді знову таки HTML-сторінки, яка може сильно відрізнятись від тієї, що зберігається на сервері. Звичайні ж сторінки, що мають розширення .html/.htm Web-сервер буде відправляти браузеру без будь-якої обробки. Основна відмінність від CGI-скриптів, написаних на інших мовах, типу Perl або C – це те, що в CGI-програмах розробник самостійно пише HTML-код, що виводиться, а, використовуючи PHP – вбудовує свою програму-скрипт в готову HTML-сторінку, використовуючи відкриваючий і закриваючий теги (<?php та ?>). PHP називається мовою серверних скриптів на відміну від Javascript/jscript/vbscript, які є мовами клієнтських скриптів. Це означає, що PHP-скрипт виконується на

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

сервері, а клієнтові передається результат його роботи, тоді як в JavaScript , код повністю передається на клієнтську машину і лише там виконується браузером.

При написанні на інших мовах програмування, наприклад, на Perl або C – замість того, щоб створювати програму, яка займається формуванням HTML-коду і містить незліченну безліч призначених для цього команд, PHP-програміст має можливість створювати HTML-код з декількома упровадженими командами PHP. Код PHP відділяється спеціальними початковим і кінцевим тегами, які дозволяють процесору PHP визначати початок і кінець ділянки HTML-коду, що містить PHP-скрипт.

Значною відмінністю PHP від якого-небудь коду, що виконується на стороні клієнта, наприклад, Javascript, є те, що PHP-скрипти виконуються на сервері. PHP-скрипт розміщений на сервері, клієнт отримує тільки результат виконання скрипта, причому клієнт не має можливості з'ясувати, який саме код виконується. Існує також можливість конфігурувати власний сервер так, щоб HTML-файли оброблялися процесором PHP, так що клієнти навіть не в змозі дізнатися, чи отримують вони звичайний HTML-файл, чи це результат виконання скрипта.

Мова програмування PHP досить проста для освоєння, але разом з тим здатна задовольнити запити професійних програмістів. Хоча PHP, головним чином, призначений для роботи в середовищі web-серверів, область його застосування не обмежується тільки цим.

Можливості мови PHP дуже великі. Головним чином, область застосування PHP сфокусована на написання скриптів, що працюють на стороні сервера; таким чином, PHP здатний виконувати все те, що виконує будь-яка інша програма CGI, наприклад, обробляти дані форм, генерувати динамічні сторінки або посилати і приймати cookies. Окрім вищенаведеного PHP здатний виконувати і безліч інших завдань.

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

Існують три основні області, де використовується PHP.

1. Створення скриптів для виконання на стороні сервера. Саме таким чином мова програмування PHP найширше використовується. Все, що може знадобитися програмісту – це парсер PHP (у вигляді програми CGI або серверного модуля), вебсервер і браузер. Для того, щоб існувала можливість проглядати результати виконання PHP-скриптів у браузері, потрібен працюючий веб-сервер і встановлений PHP.

2. Створення скриптів для виконання в командному рядку. Існує можливість створити PHP-скрипт, здатний запускатися незалежно від веб-серверу та браузера. Все, що потрібно – це парсер PHP. Такий спосіб використання PHP ідеально підходить для скриптів, які повинні виконуватися регулярно, наприклад, за допомогою cron (на платформах Unix або Linux) або за допомогою планувальника завдань (Task Scheduler) на платформах Windows. Ці скрипти також можуть бути використані в завданнях простої обробки даних.

Існує величезна кількість документації і списків розсилки, до яких можна звернутися у разі виникнення яких-небудь питань.

Perl – високорівнева, інтерпретована, динамічна мова програмування загального призначення. Perl запозичує можливості з багатьох інших мов програмування, як то C, shell scripting, AWK та sed. Мова надає потужні можливості для обробки тексту без довільних обмежень на довжину даних багатьох сучасних інструментів Unix, полегшуючи процес маніпуляції текстових файлів. Використовується для програмування графіки, системного адміністрування, у мережному програмуванні, у написанні програмного забезпечення, яке взаємодіє з базами даних, у програмуванні CGI для веб. Perl за свою гнучкість і потужність отримав прізвисько «швейцарського армійського ножа мов програмування».

Perl – мова програмування загального призначення, котра на початку розроблялась, як інструмент для обробки тексту, і тепер використовується для вирішення дуже широкого кола завдань, включно із системним

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

адмініструванням, веб-розробкою, розробкою мережного програмного забезпечення, та програмного забезпечення з графічним інтерфейсом користувача.

Мова віддає перевагу практичному (простота у використанні, ефективність, повнота) над гарним (крихітність, елегантність, мінімалістичність). Вона має багато можливостей, включно із підтримкою кількох парадигм програмування(процедурне програмування, об'єктно-орієнтоване програмування, а також функціональне програмування), управлінням пам'яттю, вбудованою підтримкою системи обробки текстів, і величезною колекцією сторонніх модулів.

Загальна структура програм на мові Perl була запозичена у C. Perl є типовою процедурною мовою, зі змінними, виразами, присвоюваннями, дужками-розділювачами, блоками, циклами та підпрограмами.

Perl також запозичив деякі особливості з shell-програмування. Назви усіх змінних розпочинаються зі спеціальних знаків, кожен з яких характеризує певний тип даних (наприклад, скаляри, масиви та хеш-таблиці). Важливо зазначити, що це дозволяє правильно інтерпретувати назви змінних безпосередньо у текстових рядках. Perl має багато вбудованих функцій, які забезпечують програміста інструментарієм, котрий часто використовується у shell-програмуванні (хоча, багато з цих інструментів реалізовується за допомогою зовнішніх інструментів), як то сортування або виклик зовнішніх програм.

Perl запозичив списки з мови Lisp, асоціативні масиви (хеші) з AWK, і регулярні вирази з sed. Це спростило процес парсингу і обробки великих обсягів текстової інформації.

У Perl 5, була додана підтримка комплексних структур даних, функції першого класу, і об'єктно-орієнтованої моделі програмування. Що включає у себе посилання, пакунки, та інше. Серед найбільших нововведень у Perl 5 слід відмітити наявність модулів, які дозволяють повторно використовувати один і той же код.

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

Усі версії Perl здійснюють автоматичне типізування змінних і управління пам'яттю. Інтерпретатор знає про тип та способи зберігання усіх об'єктів даних у програмі.

Дизайн Perl можна описати, як відповідь на три основні тенденції в комп'ютерній індустрії: зниження вартості апаратного забезпечення, зростання вартості робочої сили, а також вдосконалення технології компіляторів. Багато ранніх мов програмування, як то Fortran та C, були розроблені таким чином, щоб якомога ефективніше використовувати на той час дороге апаратне забезпечення. У протилежність, Perl було розроблено для підвищення ефективності роботи дорогих в наш час програмістів.

Perl має багато можливостей, які збільшують ефективність програміста за рахунок інтенсивного використання мікропроцесора та великих обсягів оперативної пам'яті. Серед них: автоматичне управління пам'яттю; динамічна типізація; стрічки, списки, та хеші; регулярні вирази; самоаналіз; та функція eval().

Perl підтримує мовні конструкції, які є короткими і природними для людей при їхньому читанні і написанні, навіть якщо це ускладнює реалізацію інтерпретатора Perl.

Синтаксис Perl має багато спільного з синтаксисом мов Cі, Awk, Sed і Shell. Перший рядок початкового коду може починатися з «#!/Шлях/до/Perl [-ключі]» – що вказує системі шлях до інтерпретатора Perl для виконання програми в *NIX системах і виконання їх на Веб-сервері.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи надання доступу до мережі Internet сервіс провайдера.

					VKPM-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

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-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Інтернет-провайдери підключені до однієї або кількох ліній високошвидкісного Інтернету. Більші інтернет-провайдери мають власні високошвидкісні виділені лінії, тому вони менше залежать від телекомунікаційних послуг і можуть надавати кращі послуги своїм клієнтам.

Інтернет-провайдери також зберігають тисячі серверів у центрах обробки даних – кількість серверів залежить від зони обслуговування Інтернету. Ці великі центри обробки даних керують усім трафіком клієнтів. Кілька провайдерів також підключені до великих магістральних центрів маршрутизації.

Інтернет-провайдери згруповані в три рівні:

– **Інтернет-провайдери рівня 1.** Ці інтернет-провайдери мають найбільше глобальне охоплення та володіють достатньою кількістю фізичних ліній мережі, щоб передавати більшість трафіку самостійно. Вони також домовляються з іншими мережами рівня 1, щоб дозволити безкоштовному трафіку проходити до інших провайдерів рівня 1. Інтернет-провайдери рівня 1 зазвичай продають доступ до мережі Інтернет-провайдерам рівня 2.

– **Інтернет-провайдери рівня 2.** Ці провайдери мають регіональне або національне охоплення та є постачальниками послуг, які з'єднують провайдерів рівня 1 і рівня 3. Вони повинні придбати доступ до більших мереж рівня 1, але є рівноправними з іншими провайдерами рівня 2. Мережі рівня 2 зосереджені на споживачах і комерційних клієнтах.

– **Інтернет-провайдери рівня 3.** Ці провайдери підключають клієнтів до Інтернету за допомогою мережі іншого провайдера. Інтернет-провайдери рівня 3 використовують і платять провайдерам вищого рівня за доступ до послуг

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

Як вибрати провайдера

Користувачі повинні вибрати провайдера на основі кількох факторів, зокрема таких:

– **Зона покриття.** Які провайдери пропонують послуги в регіоні користувача? Якщо користувач проживає в сільській місцевості, варіанти можуть бути обмежені.

– **Види пропонованих послуг.** Окрім кабельного, оптоволоконного, DSL чи супутникового зв'язку, чи пропонує інтернет-провайдер безпеку в Інтернеті? Безкоштовний доступ до електронної пошти? Хостинг для сайтів? Як щодо сітчастого Wi-Fi? Переконайтеся, що пропозиції інтернет-провайдера відповідають потребам користувача.

– **Швидкість завантаження та завантаження.** Чи буде користувач грати в онлайн-ігри чи працювати вдома та використовувати відеотелеконференції? Обидва потребують різного рівня обслуговування. Наприклад, для потокової передачі відео 4K потрібна пропускна здатність не менше 25 Мбіт/с.

– **Ціноутворення.** Чи поєднує провайдер таких послуг, як Інтернет, телефон і телебачення, і якщо так, чи економить об'єднання послуг? Чи є якісь обмеження даних? Що щодо вартості обладнання? Чи є договір?

– **Рейтинг задоволеності споживачів.** Перевірте рейтинги постачальників в неупереджених джерелах.

Вимоги до пропускної здатності для різних мультимедійних і голосових функцій.

Інтернет-провайдери можуть зменшувати або сповільнювати швидкість Інтернету користувача, щоб регулювати трафік і усунути перевантаження мережі. Інтернет-провайдери також можуть зменшувати швидкість Інтернету користувача, коли користувач досягає певного ліміту даних. Однак обмеження порушує ідею мережевого нейтралітету, яка є переважаючою думкою про те, що провайдери мають однаково ставитися до всіх комунікацій через Інтернет.

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

Наприклад, інтернет-провайдери можуть гальмувати певні веб-сайти, які відвідують користувачі, просто тому, що вони займають багато даних. У минулому інтернет-провайдери обмежували доступ до Інтернету своїх клієнтів під час підключення до Netflix, тобто через інтернет-провайдера користувальницький досвід роботи на платформі Netflix погіршувався.

Необхідно визначити, для яких цілей необхідний інтернет. Від цього залежить саме головне – вибір швидкості з'єднання. Робота з поштою й документами, перегляд сайтів вимагають приблизно 8 Мбіт/с, трафік можливий лімітований. Для спілкування по Skype, онлайн-ігор, завантаження невеликих файлів потрібний безлімітний інтернет і мінімальну швидкість 25 Мбіт/с. Перегляд фільмів онлайн, активне завантаження інформації й мережні ігри забезпечує трафік на 40 Мбіт/с. Всі ці моменти необхідно враховувати при рішенні питання, як вибрати провайдера.

Необхідно також з'ясувати, який тип підключення буде найбільш підходящий: звичайний телефонний (що комутирується), використання технології ADSL, виділена лінія, бездротове підключення, мобільний інтернет. Кожний спосіб має свої плюси й мінуси, тому необхідно брати до уваги такі критерії, як вартість послуг, гарантуєма швидкість і стабільність з'єднання, наявність якісної техпідтримки й так далі. Звичайно, ідеального провайдера не існує, але оптимальне сполучення всіх критеріїв, що цікавлять, знайти цілком можливо.

За допомогою інтернету можливо обмінюватися повідомленнями, одержувати й відправляти листи, купувати й продавати товари в інтернет-магазинах, оновлювати програми, установлені на спеціальних серверах, користуватися можливостями інтернет-телебачення й радіомовлення, безкоштовно дзвонити на інший комп'ютер, спілкуватися в соціальних мережах, оплачувати рахунку й так далі. Що ж необхідно для того, щоб одержати доступ в інтернет?

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

Перше – вибрати спосіб підключення, друге – укласти інтернет-провайдером договір. У залежності про того, який спосіб підключення вибирається, знадобиться відповідне встаткування. Так, для підключення через модем ADSL (сьогодні це найпоширеніше з'єднання) потрібно властиво провідна телефонна лінія й зовнішній модем.

Підключення, що здійснюється за допомогою виділеної оптоволоконої лінії FTTB, саме надійне, що надає найвищу швидкість передачі даних і що дозволяє передавати десятки каналів IP-TV одночасно. Для того щоб підключитися до кабельної мережі, необхідні оптоволоконий кабель і спеціальний зовнішній пристрій, розташований поза будинком або квартирою й обслуговуюча певна кількість користувачів.

Низькошвидкісне підключення через мережу GPRS – вид бездротового підключення, для якого потрібний тільки мобільний телефон.

Бездротове високошвидкісне з'єднання 3G, дуже популярне сьогодні в тих регіонах, де немає ще кабельного інтернету, забезпечує підключення через стільникову мережу. У цьому випадку встаткування для Інтернету являє собою наявність спеціального 3G-модему.

Wi-Fi підключення, також бездротове, вимагає не тільки наявності Wi-Fi зони, але й спеціально передбаченої відповідної опції комп'ютера користувача. Такі зони створюють великі провайдери на окремих територіях.

Підключення через електромережу із придбанням спеціального модему мало поширено, хоча забезпечує досить швидке з'єднання. При цьому в будинку повинне бути встановлене спеціальне устаткування.

Деякі провайдери займаються продажем трафіку або часу, проведеного в інтернеті. Підключення здійснюється по локальній мережі через проксі-сервер з використанням модему. Потрібно купити спеціальну картку й підключатися або по логіну й паролі, або по IP-адресі й порту до інтернет-сервера.

З'єднанням через супутник найменше користуються в містах, але воно незамінно в приміській зоні. Щоб скористатися таким видом підключення до інтернету, потрібна супутникова тарілка й відповідне спеціальне устаткування.

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

Маршрутизатори в глобальних мережах

Маршрутизатори для глобальних мереж – це ті, за допомогою яких компанії можуть вирішити завдання переміщення даних між віддаленими офісами. Вони призначені для глобальних мереж, а не для Internet. Вони переміщують пакети, але не є комутаторами третього рівня. Маршрутизатори для глобальних мереж – це те, що можливо використовувати для організації зв'язку між Нью-Йорком, Лондоном і Києвом, або Кіровоградом.

При виборі маршрутизатора для глобальної мережі першочергова увага варто звертати на такі фактори, як надійність, керованість, розширюваність, гнучкість, спектр пропонованих технологій і підтримувані протоколи.

Як правило, кожний виробник сповідає свій власний підхід до конфігурації маршрутизаторів. Це стосується розходжень не тільки в командах і графічному інтерфейсі, але й у самій концепції й термінології.

Маршрутизатори відомі труднощами конфігурації. На щастя, постачальники домоглися великого прогресу принаймні в найпростішому випадку – один порт локальної мережі й один послідовний порт (або лінія ISDN) до центрального офісу або в Internet. Однак у багатьох трохи більше складних топологіях локальних і глобальних мереж з декількома інтерфейсами, протоколами й алгоритмами маршрутизації конфігурувати маршрутизатори стало сутужніше, ніж усього лише рік назад. Чому? Через нові функції й можливості, кожна з яких вимагає ретельного налаштування, щоб мережа не виявилася в стані хаосу.

Рішення від одного виробника не дуже популярні серед адміністраторів мереж. Ніхто не бажає виявитися заручником планів випуску, поточних цін і підтримуємих функцій постачальника. Разом з тим витрати на навчання й підтримку продуктів від декількох постачальників – багато хто з яких роблять те саме – дуже важко зіставити із втратами через прихильність одному постачальникові, особливо якщо він не задовольняє всім вашим вимогам.

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

Це пояснює, чому найбільші виробники, у тому числі Cisco, Bay і 3Com, стають ще більше, й чому багато хто інші пішли з ринку маршрутизаторів або переорієнтувалися на маршрутизатори доступу в Internet. Якщо ви зрозуміли, як змусити працювати у вашій мережі, скажемо, маршрутизатор Bay, то навряд чи захочете проробити той же шлях із продуктами 3Com або Cisco. Тому свій вибір ви зупините в першу чергу на тім постачальнику, про який точно знаєте, що він не зникне через кілька років і запропонує необхідний спектр продуктів і функцій.

Вибір одного постачальника може виявитися проте недостатнім. Наприклад, Cisco Systems придбала настільки багато мережних компаній, що деякі з її маршрутизуючих і комутуючих продуктів мають моделі з конфігурацією, досить відмінної від прийнятої в Internetworking Operating System (IOS). У загальному ж випадку робота з одним виробником позбавить вас від чималого числа проблем при конфігурації й налаштуванні.

Уніфікація

Незважаючи на те що усі, зібралися підключатися до Internet (тобто використовувати TCP/IP), адміністратори корпоративних мереж знають, що цим їхні турботи далеко не вичерпуються. Компанії самих різних розмірів працюють і з іншими протоколами (як застарілими, так і немає) для мереж Apple, Digital, IBM і Novell. Щоб мережа працювала, ці протоколи потрібно маршрутизувати.

Звичайно виробники пропонують один комплект програмного забезпечення, загальний для всіх його апаратних платформ. Придбавши програмне забезпечення, ви одержуєте ту саму функціональність для встаткування як молодшого, так і старшого класу. Крім того, якщо виробник підтримує широкий спектр протоколів, то він часто пропонує програмне забезпечення відповідно до набору функцій: базові версії для підтримки декількох протоколів локальних мереж, глобальних мереж і маршрутизації; версії середнього й старшого класу з більше складними функціями й протоколами; спеціальні версії для специфічних завдань, наприклад для брандмауерів, віртуальних мереж і шифрування.

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

При виборі лінії продуктів для глобальної мережі необхідно переконатися, що необхідні функції доступні у всьому спектрі встаткування, що ви збираєтеся використовувати. У деяких постачальників, зокрема Digital, Compatible Systems і Bay Networks, програмні функції відрізняються кардинальним образом при переході від устаткування молодшого класу до встаткування старшого класу. Якщо ви зупинили свій вибір на одному зі згаданих постачальників, то я радив би перевірити, що необхідні функції підтримуються всім устаткуванням, яким ви плануєте користуватися.

Всі розглянуті виробники підтримують основні протоколи для локальних мереж: TCP/IP, IPX, AppleTalk і DECnet Phase IV. У той же час їхнього виробу можуть організувати міст для немаршрутизуємих протоколів типу NetBEUI (цей протокол використовується в мережах LAN Manager, у тому числі Windows NT) і LAT (протокол сервера терміналів розробки Digital).

Якщо глянути, хто що ще пропонує, то ми знайдемо масу відмінностей. Наприклад, 3Com, Cisco, Digital, IBM і Xyrplex просувають повну маршрутизацію у відповідності зі стандартами OSI, у той час як підтримка SNA і більше старих мережних протоколів IBM є сильною стороною продуктів від 3Com, Bay Networks, Cisco, IBM і TimePlex.

Зі старими протоколами, такими, як OSI і SNA, працювати досить просто, тому що стандарти давно не міняються, а реалізації добре налагоджені. У подібному середовищі замовникові досить надати виробникові список вимог, щоб останній підтвердив можливість підтримки його мережі.

Що стосується активно розроблювальних протоколів, особливо TCP/IP, справи йдуть набагато складніше. Наприклад, IPv6, що впливає версія протоколу мережного рівня в стеці TCP/IP, активно просувається 3Com, Bay Networks і групою мережних продуктів Digital. Підтримка різних нових розширень для IP міняється з кожним днем і від продукту до продукту, тому я підготував список функцій, про підтримку яких можливо поцікавитися у свого постачальника.

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

з'єднань T-1. Звичайно високошвидкісні з'єднання обмежені винятково передачею IP-пакетів. Навіть якщо програмне забезпечення виробника підтримує інші протоколи, то можливо зштовхнутися з тим, що маршрутизатор виявиться не в змозі забезпечувати настільки високі швидкості, якщо від нього буде потрібно надто багато.

За винятком "найбільших з великих", корпоративним мережам звичайно досить з'єднань DS3 (близько 45 Мбіт/с) і менш. Однак не всі DS3 однакові. Деякі оператори далекого зв'язку пропонують чистий канал DS3, всі передані по якому від краю до краю біти – ваші. Такий канал надзвичайно дорогий, тому що по суті оператор виділяє еквівалент 28 каналів T-1 одному замовникові. Звичайно ж оператори надають DS3 у вигляді ATM.

ATM володіє рядом переваг: він дозволяє зв'язати кілька вузлів, причому кожний з них може мати своє високошвидкісне з'єднання. У випадку ATM, однак, багато бітів використовуються непродуктивно. Наприклад, для DS3 на 45 Мбіт/с ATM забезпечує пропускну здатність приблизно в 34 Мбіт/с.

Вибір між чистим каналом і ATM має немаловажні наслідки: далеко не всі маршрутизатори підтримують обидві опції. Наприклад, VSR від Compatible Systems може передавати пакети PPP або SMDS через інтерфейс DS3, але не ATM. VSR, однак, має одна перевага над конкурентами: він містить убудований T-3 DSU/CSU. Як виявилися, лише деякі маршрутизатори мають аналогічну опцію. Її наявність дозволяє заощадити на покупці, конфігурації й керуванні досить дорогим компонентом устаткування.

Відсутність убудованого DSU/CSU пояснює також наявність інтерфейсів HSSI на всіх розглянутих продуктах (за винятком Xyplex Network 900 і "річкової" серії ACC). Сигнал DS3 повинен якимось образом бути доставлений від DSU/CSU до маршрутизатора, а стандартний послідовний кабель не здатний зробити це настільки надійно, як у випадку T-1. Разом з тим при швидкостях вище DS3 оптичне з'єднання завершується безпосередньо на маршрутизаторі.

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

Якщо у вас установлені мейнфрейми IBM, то може знадобитися розглянути ще один інтерфейс: з'єднання для корпоративних систем (Enterprise Systems Connection, ESCON) і паралельні канали, які підтримують тільки пристрої Cisco і IBM.

Види роутерів

Роутер – спеціалізований мережний комп'ютер, що має як мінімум один мережний інтерфейс і пересилаючий пакети даних між різними сегментами мережі, що зв'язує різнорідні мережі різних архітектур. Дозволяє обмінюватися даними між підключеними до нього пристроями, а так само надає їм загальний доступ в Інтернет. Якщо говорити простіше, те роутер «одержує» інтернет тим або іншом способом, і «роздає» його підключеним пристроям: комп'ютеру, ноутбуку, планшету, смартфону й т.д. Отже, як же вибрати роутер, адже на ринку зараз кілька сотень моделей, кожна з яких дійсно відрізняється від інших. Щоб відповісти на це питання, необхідно визначитися: для чого вам потрібний роутер – для особистого користування вдома або для роботи (офісу)? Саме від цього будуть залежати ключові показники пристрою, по яких вам доведеться вибрати необхідну модель.

3.2 Розробка структурної схеми

Для розуміння як саме реалізовано у дипломі система надання доступу до мережі Інтернет сервіс провайдера розглянемо основу мережі, а саме TCP/IP.

Протокол керування передачею (TCP) – це стандарт зв'язку, який дозволяє прикладним програмам і комп'ютерним пристроям обмінюватися повідомленнями через мережу. Він призначений для надсилання пакетів через Інтернет і забезпечення успішної доставки даних і повідомлень через мережі.

TCP є одним із основних стандартів, які визначають правила Інтернету, і входить до стандартів, визначених Інженерною робочою групою Інтернету (IETF). Це один із найбільш часто використовуваних протоколів у цифровій

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

мережі зв'язку, який забезпечує наскрізну доставку даних.

TCP організовує дані таким чином, щоб їх можна було передавати між сервером і клієнтом. Це гарантує цілісність даних, які передаються через мережу. Перш ніж передати дані, TCP встановлює з'єднання між джерелом і одержувачем, яке, як він гарантує, залишається активним до початку зв'язку. Потім він розбиває великі обсяги даних на менші пакети, забезпечуючи при цьому цілісність даних протягом усього процесу.

У результаті всі протоколи високого рівня, яким потрібно передавати дані, використовують протокол TCP. Приклади включають однорангові методи спільного використання, як-от протокол передачі файлів (FTP), Secure Shell (SSH) і Telnet. Він також використовується для надсилання й отримання електронної пошти через протокол доступу до повідомлень в Інтернеті (IMAP), протокол поштового відділення (POP) і простий протокол передачі пошти (SMTP), а також для доступу до Інтернету через протокол передавання гіпертексту (HTTP).

Альтернативою TCP у мережі є протокол датаграм користувача (UDP), який використовується для встановлення з'єднань із малою затримкою між програмами та зменшення часу передачі. TCP може бути дорогим мережевим інструментом, оскільки він містить відсутні або пошкоджені пакети та захищає доставку даних за допомогою таких елементів керування, як підтвердження, запуск з'єднання та керування потоком.

UDP не забезпечує підключення через помилку чи послідовність пакетів, а також не сигналізує про призначення перед доставкою даних, що робить його менш надійним, але менш дорогим. Таким чином, це хороший варіант для чутливих до часу ситуацій, таких як пошук у системі доменних імен (DNS), голос через Інтернет-протокол (VoIP) і потокове медіа.

Інтернет-протокол (IP) – це спосіб передачі даних з одного пристрою на інший через Інтернет. Кожен пристрій має IP-адресу, яка унікально ідентифікує його та дозволяє йому спілкуватися та обмінюватися даними з іншими пристроями, підключеними до Інтернету. Сьогодні це вважається стандартом

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

швидкого та безпечного зв'язку безпосередньо між мобільними пристроями.

IP відповідає за визначення того, як програми та пристрої обмінюються пакетами даних один з одним. Це основний протокол зв'язку, який відповідає за формати та правила обміну даними та повідомленнями між комп'ютерами в одній мережі або кількох підключених до Інтернету мережах. Це робиться за допомогою набору протоколів Інтернету (TCP/IP), групи протоколів зв'язку, які розділені на чотири рівні абстракції.

IP є основним протоколом на рівні Інтернету TCP/IP. Його основна мета полягає в доставці пакетів даних між вихідною програмою або пристроєм і одержувачем за допомогою методів і структур, які розміщують теги, такі як адресна інформація, в пакетах даних.

TCP та IP – це окремі протоколи, які працюють разом, щоб гарантувати доставку даних до місця призначення в мережі. IP отримує та визначає адресу – IP-адресу – програми або пристрою, на які мають бути надіслані дані. Потім TCP відповідає за транспортування та маршрутизацію даних через мережеву архітектуру та забезпечення їх доставки до цільової програми або пристрою, визначеного IP. Обидві технології, що працюють разом, забезпечують зв'язок між пристроями на великих відстанях, дозволяючи передавати дані туди, куди потрібно, найефективнішим способом.

Іншими словами, IP-адреса схожа на номер телефону, призначений смартфону. TCP – це комп'ютерна мережева версія технології, яка використовується для того, щоб смартфон дзвонив і дозволяв його користувачеві розмовляти з особою, яка йому телефонувала.

Тепер, коли ми розглянули TCP та IP окремо, що таке TCP/IP? Два протоколи часто використовуються разом і залежать один від одного, щоб дані мали місце призначення та безпечно досягали його, тому цей процес регулярно називають TCP/IP. Завдяки належним протоколам безпеки комбінація TCP/IP дозволяє користувачам виконувати безпечний процес, коли їм потрібно переміщувати дані між двома або більше пристроями.

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

Як працює протокол керування передачею (ТСР)/ІР?

Модель ТСР/ІР є типовим методом передачі даних в Інтернеті. Він був розроблений Міністерством оборони Сполучених Штатів для забезпечення точної та правильної передачі даних між пристроями. Він розбиває повідомлення на пакети, щоб уникнути повторного надсилання всього повідомлення, якщо під час передачі виникне проблема. Пакети автоматично збираються, коли вони досягають місця призначення. Кожен пакет може пройти різним маршрутом між комп'ютером джерела та одержувачем, залежно від того, чи стає вихідний маршрут перевантаженим або недоступним.

ТСР/ІР поділяє комунікаційні завдання на рівні, які зберігають процес стандартизованим, без постачальників обладнання та програмного забезпечення, які здійснюють керування самостійно. Пакети даних мають пройти через чотири рівні, перш ніж їх отримає пристрій призначення, потім ТСР/ІР проходить через рівні у зворотному порядку, щоб повернути повідомлення у вихідний формат.

Як протокол на основі з'єднання, ТСР встановлює та підтримує з'єднання між програмами або пристроями, доки вони не завершать обмін даними. Він визначає, як вихідне повідомлення має бути розбито на пакети, нумерує та збирає пакети, а потім надсилає їх на інші пристрої в мережі, такі як маршрутизатори, шлюзи безпеки та комутатори, а потім до місця призначення. ТСР також надсилає та отримує пакети з мережевого рівня, обробляє передачу будь-яких відкинутих пакетів, керує потоком і гарантує, що всі пакети досягнуть місця призначення.

Хорошим прикладом того, як це працює на практиці, є надсилання електронного листа за допомогою SMTP із сервера електронної пошти. Щоб розпочати процес, рівень ТСР на сервері ділить повідомлення на пакети, нумерує їх і пересилає на рівень ІР, який потім транспортує кожен пакет на сервер електронної пошти призначення. Коли пакети надходять, вони повертаються на рівень ТСР, щоб повторно зібратися у вихідний формат повідомлення та передаються назад на сервер електронної пошти, який доставляє повідомлення до

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

скриньки електронної пошти користувача.

TCP/IP використовує тристороннє рукостискання для встановлення з'єднання між пристроєм і сервером, що забезпечує одночасну передачу кількох з'єднань TCP-сокетів в обох напрямках. І пристрій, і сервер мають синхронізувати та підтверджувати пакети перед початком зв'язку, тоді вони зможуть погоджувати, розділяти та передавати з'єднання сокетів TCP.

4 рівні моделі TCP/IP

Модель TCP/IP визначає, як пристрої мають передавати дані між собою, і забезпечує зв'язок через мережі та великі відстані. Модель представляє, як дані обмінюються та організовуються в мережах. Він розділений на чотири рівні, які встановлюють стандарти для обміну даними та представляють, як дані обробляються та упаковуються під час доставки між програмами, пристроями та серверами.

Чотири рівні моделі TCP/IP такі:

1. Канальний рівень: Канальний рівень визначає спосіб надсилання даних, обробляє фізичний акт надсилання й отримання даних і відповідає за передачу даних між програмами чи пристроями в мережі. Це включає визначення того, як дані повинні сигналізувати апаратне забезпечення та інші пристрої передачі в мережі, такі як драйвер пристрою комп'ютера, кабель Ethernet, карта мережевого інтерфейсу (NIC) або бездротова мережа. Його також називають канальним рівнем, рівнем доступу до мережі, рівнем мережевого інтерфейсу або фізичним рівнем і є комбінацією фізичного рівня та рівня каналу даних моделі взаємозв'язку відкритих систем (OSI), яка стандартизує функції зв'язку в обчислювальних і телекомунікаційних системах. системи.

2. Інтернет-рівень: Інтернет-рівень відповідає за надсилання пакетів із мережі та контроль їх переміщення через мережу, щоб гарантувати, що вони досягнуть місця призначення. Він надає функції та процедури для передачі послідовностей даних між програмами та пристроями через мережі.

3. Транспортний рівень: транспортний рівень відповідає за забезпечення

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

міцного та надійного з'єднання даних між початковою програмою чи пристроєм і його призначенням. Це рівень, на якому дані поділяються на пакети та нумеруються для створення послідовності. Потім транспортний рівень визначає, скільки даних потрібно надіслати, куди їх слід надіслати та з якою швидкістю. Він гарантує, що пакети даних надсилаються без помилок і послідовно, і отримує підтвердження того, що пристрій призначення отримав пакети даних.

4. Прикладний рівень: прикладний рівень відноситься до програм, яким потрібен TCP/IP, щоб допомогти їм спілкуватися одна з одною. Це рівень, з яким зазвичай взаємодіють користувачі, як-от системи електронної пошти та платформи обміну повідомленнями. Він поєднує сеансовий, презентаційний і прикладний рівні моделі OSI.

Ваші пакети даних приватні через TCP/IP?

Пакети даних, надіслані через TCP/IP, не є приватними, а це означає, що їх можна побачити або перехопити. З цієї причини життєво важливо уникати використання загальнодоступних мереж Wi-Fi для надсилання особистих даних і переконатися, що інформація зашифрована. Одним із способів шифрування даних, які надаються через TCP/IP, є віртуальна приватна мережа (VPN).

Що таке моя TCP/IP-адреса?

TCP/IP-адреса може знадобитися для налаштування мережі та, швидше за все, потрібна в локальній мережі.

Пошук загальнодоступної IP-адреси – це простий процес, який можна знайти за допомогою різних онлайн-інструментів. Ці інструменти швидко визначають IP-адресу використовуваного пристрою, а також IP-адресу хоста користувача, Інтернет-провайдера (ISP), віддалений порт і тип браузера, пристрою та операційної системи, які вони використовують.

Інший спосіб виявити TCP/IP – через сторінку адміністрування маршрутизатора, яка відображає поточну публічну IP-адресу користувача, IP-адресу маршрутизатора, маску підмережі та іншу інформацію про мережу.

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

На рисунку 3.1 зображена структурна схема системи. На який детально розглянута розроблена система. Розглянемо роботу розробленої системи на прикладі запити Інтернету користувача №N.

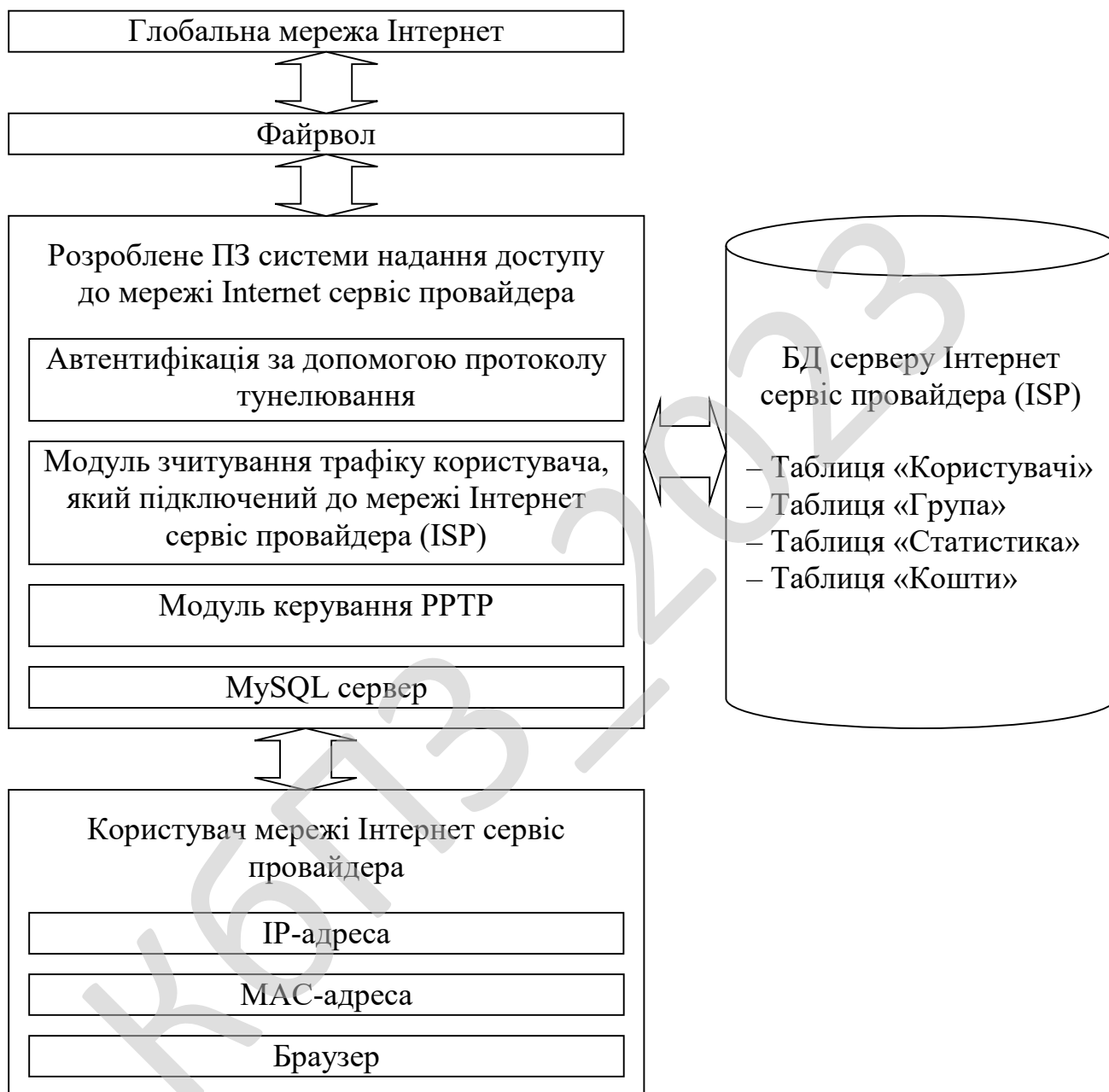


Рисунок 3.1 – Структурна схема системи

Користувач N через мережу через WEB браузер робочої станції посилає запит до мережі Інтернет сервіс провайдера (ISP) з даними IP адреси. На сервері

розроблене ПЗ проводить автентифікацію робочої станції за допомогою протоколу тунелювання та далі проведе запит сторінки через брандмауер сервера. В цей час розроблене ПЗ проведе зчитування трафіку мережі, через РРТР, керування відбувається через відповідний модуль. Всі дані роботи користувача у Інтернеті заносяться до бази даних сервера ISP. Основні таблиці це «Таблиця Кошти» де зберігаються дані витрачених коштів, «Таблиця Користувачі» де знаходиться інформація користувача, «Таблиця Статистика» статистичні дані та «Таблиця Група» де зберігаються встановлені привілеї користувача.

Адміністратор ISP за допомогою розробленого магістерського ПЗ проведе моніторинг мережі. При необхідності переглядає базу даних. Основне призначення системи це моніторинг трафіку користувачів, які підключені до ISP з можливістю детального перегляду статистики роботи користувачів, розрахунок коштів, інформацію Інтернет серфінгу користувача та ін.

3.3 Розробка функціональної схеми

На рисунку 3.2 зображена функціональна схема системи. Нижче розглянемо її більш докладно.

Існують наступні функціональні можливості:

- Облік НТТР– трафіку (керована маршрутизація до проксі-сервера).
- Тарифний облік можливий у двох варіантах: обмеження використання трафіку в годину при оплаті за часом і оплата за фактом використання трафіку й витраченого часу.
- Підтримка гнучкої цінової політики. Можна встановити незалежну ціну на кожний комп'ютер, у т.ч. залежно від часу доби. Підтримка погодинних і пакетних тарифів (у т.ч. "нічний режим"). Підтримка автоматичних знижок по днях тижня й часу доби. Можливість дати клієнтові безкоштовні додаткові 5 хвилин у випадку апаратних проблем (зберігається в журналі, налаштовується адміністратором).

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42



Рисунок 3.2 – Функціональна схема системи

- Підтримка передоплати й оплати по факту. Можливість створити предоплачений сеанс як на певний час, так на задану суму.
- Можливість "розбивки" залишку засобів по різних комп'ютерах.
- Ведення каси з повним журналом подій. Збереження історії й статистики по кожному менеджері (адміністраторові). Підбиття статистики по кожному окремо взятому комп'ютеру.
- Підтримка дисконтів (у т.ч. обмежених за часом) для постійних клієнтів.
- Можливість задати індивідуальну цінову політику для різних комп'ютерів (у т.ч. виділити VIP і low-end комп'ютери), а також залежно від часу доби.
- Висока стійкість до злому.
- Можливість очистити всі завдання, запущені клієнтом (у т.ч. "завислі" ігри).
- Вбудовані модулі складу й продажів (облік супутніх товарів і послуг Cd-диски, напої і т.д., актуальних для Інтернет-кафе).
- Журналювання всіх входів у систему адміністраторів і менеджерів ISP.
- Можливість "заморожування" сеансу зі збереженням даних користувача й повного "заморожування" його програм.
- "Очищення" сеансу від завислих процесів з комп'ютера менеджера.
- Авто очищення системи ISP від несистемних процесів між сеансами.
- Можливість відправлення повідомлень користувачеві (відображаються в TRAY), робити тихіше звук на його комп'ютері, штрафувати за непристойну поведінку.
- Автоматичне завершення небажаних і небезпечних процесів ISP (список налаштовується адміністратором Інтернет-кафе / комп'ютерного клубу).
- Докладне журналювання всіх дій адміністраторів (менеджерів) для хазяїна Інтернет-кафе / комп'ютерного клубу.
- Розблокування комп'ютерів ISP з використанням подвійного коду безпеки (пароля) у випадку порушення зв'язку із сервером.

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

– Звукові оповіщення про порушення працездатності локальної мережі або зв'язку із сервером.

– Сеанси відновлюються навіть при збої/перезавантаженню сервера або клієнта. Сеанс зникне тільки в тому випадку, якщо одночасно виключити й сервер, і клієнт, але навіть у цьому випадку дані можна буде відновити вручну по журналу подій.

– Комп'ютери можуть працювати в off-line режимі (з виключеним сервером). При цьому фінансові транзакції обробляються коректно.

– За замовчуванням використовує обмежений робочий стіл Windows у якості клієнтської оболонки, що створює додаткові зручності для користувача в силу звичності.

– При бажанні можна замість робочого стола встановити будь-яку іншу оболонку сторонньої розробки.

– Висока чутливість на дії користувача, тому що зв'язок здійснюється по протоколу TCP.

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

3.4 Розробка діаграми процесів

Діаграма взаємодії процесів системи, розробленої у результаті виконання магістерського проектування, наведена на рисунку 3.3. Після початку роботи проходить автентифікація з використанням протоколу тунелювання та після завантаження демону РРТР з можливістю моніторингу демону РРТР ми потрапляємо до інтерфейсу ПЗ.

Звідки ми можемо потрапити до налаштування ПЗ, статистика роботи та моніторинг локальної мережі з доступом до БД роботи локальної мережі через MySQL сервер.

РРТР це тунельний протокол типу точка-точка, що дозволяє комп'ютеру

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

встановлювати захищене з'єднання з сервером за рахунок створення спеціального тунелю в стандартній, незахищеній мережі. PPTP поміщає (інкапсулює) кадри PPP в IP-пакели для передачі по глобальній IP-мережі, наприклад Інтернет.

PPTP може також використовуватися для організації тунелю між двома локальними мережами. PPTP використовує додаткове TCP-з'єднання для обслуговування тунелю.

Протоколи тунелювання потрібні тому, що деякі протоколи не можуть маршрутизуватися через певні мережі. Кадри PPP, які не маршрутизуються через мережу Інтернет, самостійно ці кадри не можуть знайти правильний маршрут через Інтернет до одержувача, для цього їм потрібна підтримка. Саме для цього і потрібні протоколи тунелювання. Вони дозволяють проводити інкапсуляцію пакетів PPP, тим самим кадри правильно доходять до одержувача.

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.

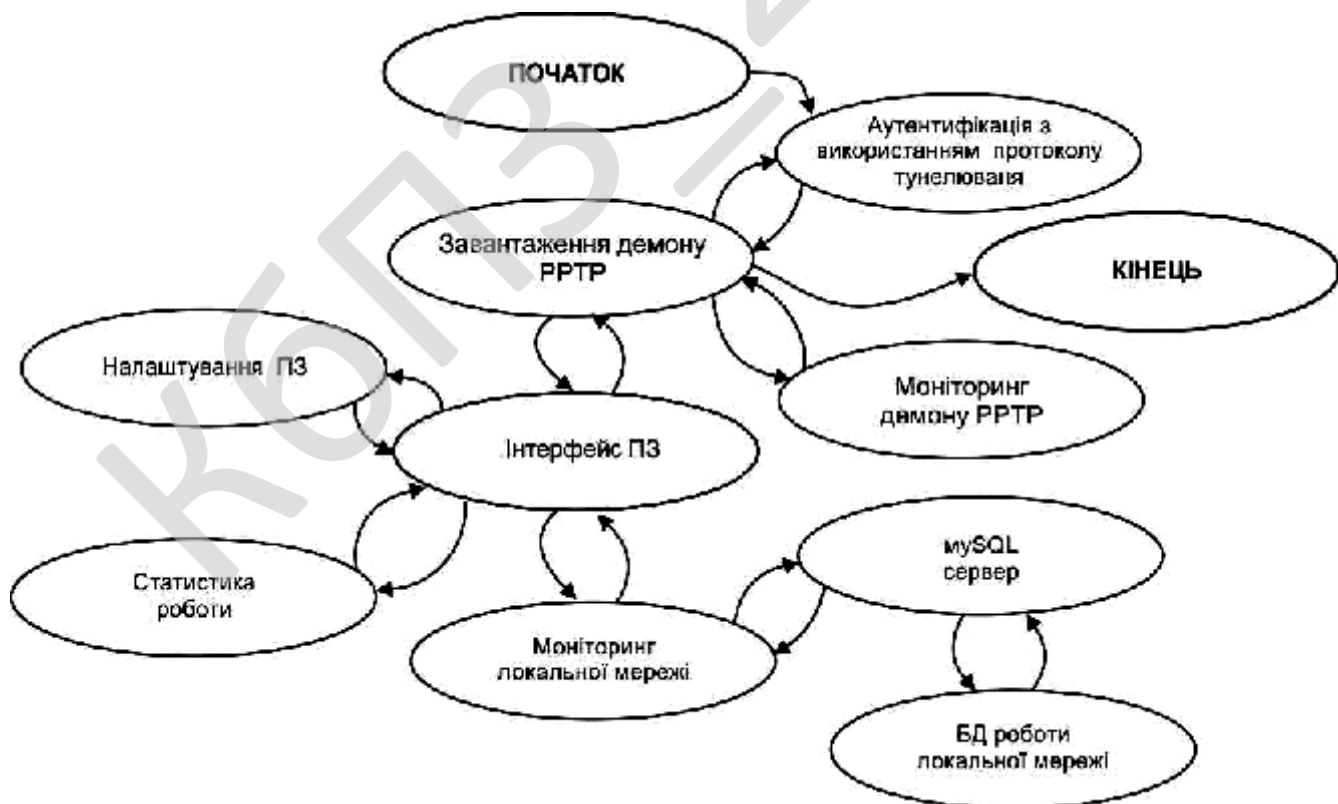


Рисунок 3.3 – Діаграма взаємодії процесів

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

На рисунку 4.1 наведено блок-схему програми. Її робота складається з виконання наступних кроків:

- Ініціалізація серверу.
- Підключення та запуск на виконання системних скриптів.
- Створення скрипкового каналу на стороні серверу.

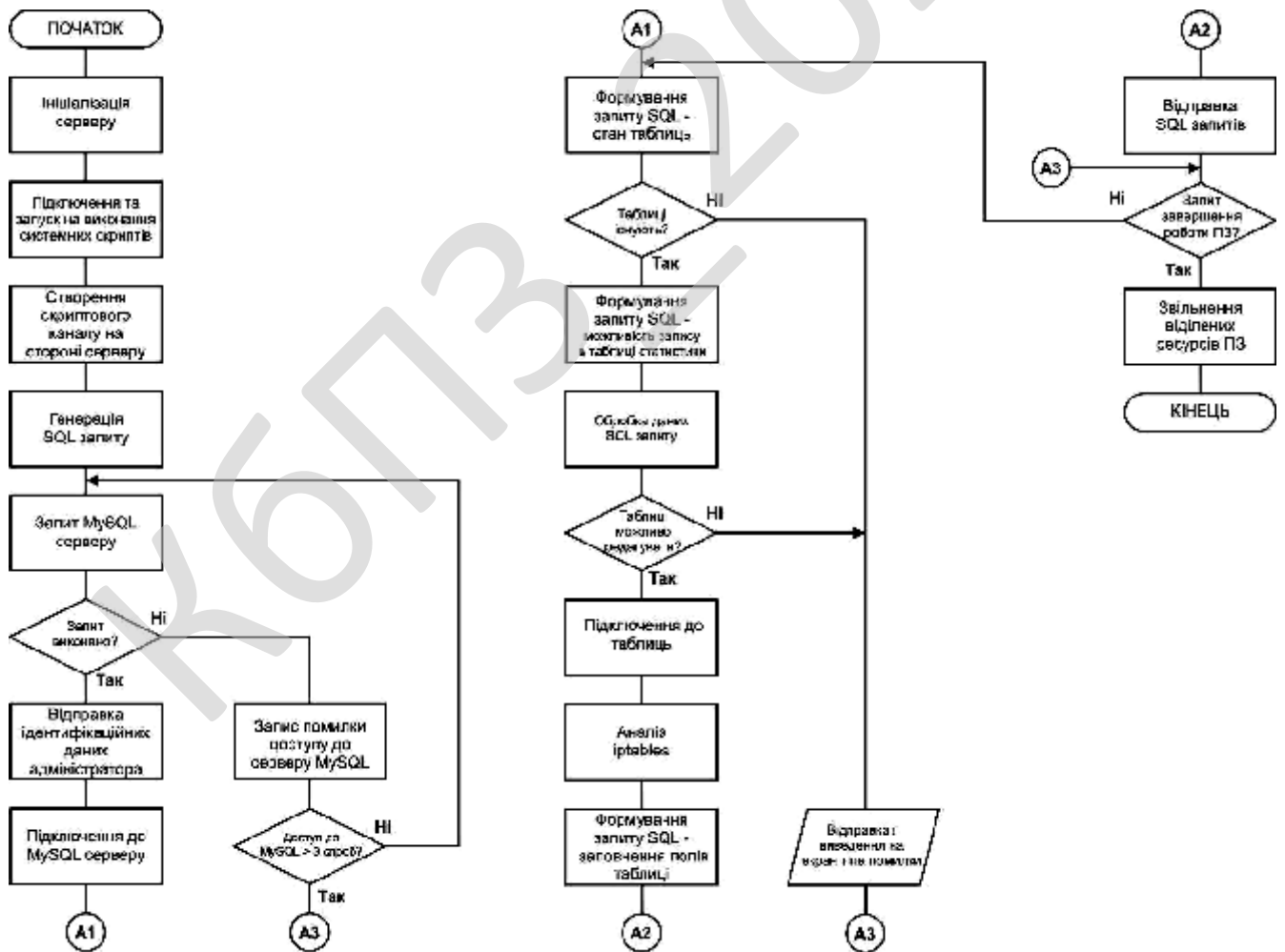


Рисунок 4.1 – Блок-схема програми

- Генерація SQL запиту.
- Запит MySQL серверу (запит).
- Відправка ідентифікаційних даних адміністратора.
- Підключення до MySQL серверу.
- Формування запиту SQL – стан таблиць (запит).
- Формування запиту SQL – можливість запису в таблиці статистики.
- Обробка даних SQL запиту.
- Запит таблиці можливо редагувати?.
- Підключення до таблиць .
- Аналіз iptables.
- Формування запиту SQL – заповнення полів таблиці.
- Відправка SQL запитів.
- Запит завершення роботи ПЗ – звільнення виділених ресурсів ПЗ.

На рисунку 4.2 наведено блок-схему інтерфейсу клієнтського ПЗ. Її робота складається з виконання наступних кроків:

- Ініціалізація клієнтського ПЗ.
- Підключення та запуск на виконання системних скриптів.
- Створення скрипкового каналу на стороні клієнта.
- Генерація SQL запиту.
- Запит MySQL серверу.
- Відправка ідентифікаційних даних користувача ПК.
- Підключення до MySQL серверу .
- Формування запиту SQL – доступ до читання таблиці статистики (запит).
- Підключення до таблиць.
- Формування запиту SQL – статистика по користувачу.
- Формування коду нижньої панелі.
- Формування коду лівої панелі.
- Формування коду верхньої панелі.
- Створення HTML коду сторінки.


```

    if (/login=(.*)/) { $login=&urldecode ($1); }
    if (/pass=(.*)/) { $password=&urldecode ($1); }
}

# Відкриваємо базу даних і перевіряємо логін і пароль.
open(INFO,$file) ||die;
@data=;
close(INFO);
foreach $string(@data) {
    @item=split(/&/,$string);
    foreach (@item) {
        if (($item[0] eq $login) && ($item[1] eq $password)) {
# Якщо все нормально, переходимо в користувацький каталог.
        print "Content-type:text/html\n\n";
        print <<HTML;
        <html><body bgcolor="e6e8fa">
# Вітаємо користувача.
        <p><h2 align=center><font color="ff0000">Hello,$login!</font></h2>
        <p><center>Welcome to your home directory!
        <p>Your URL is <a href="$basedir$login">$basedir$login.</a></center>
        HTML
# Список директорії
        $userdir=$dir.$login;
        chdir ("$userdir");
# Відкриваємо каталог і читаємо список файлів у масив.
        opendir(DIR,"$userdir") || die "Cannot open $userdir!";
        while (@files=readdir(DIR)) {
# Якщо каталог містить підкаталоги, виводимо їх
# окремо, а також не показуємо
# каталоги "." і ".." Друкуємо шапку таблиці.
        print <<HTML;
        <p><center>
        <table bgcolor="\bfbfbf\" width=600 border cellspacing=0
        cellpadding=0 nowrap>
        <tr><td colspan=5 align=center nowrap>
        <b><font color="ff0000">Directories</font></b></td></tr>
        <tr><td>.</td><td align=center><b>List</b>
        </td><td><b>Size</b>
        <td><td><b>Last accessed</b></td><td><b>Last modified</b></td>
#HTML
        foreach $file(@files) {
# Статистика файлів - розмір, час останнього звернення й модифікації.
        $size=(stat("$userdir/$file")) [7];

```

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

```

    $atime=localtime((stat("$userdir/$file"))[8]);
    $mtime=localtime((stat("$userdir/$file"))[9]);
# друкуємо список підкаталогів.
    if ( -d "$userdir/$file" && "$file" ne "." && "$file" ne "..") {
    print "<tr><td width=30><img src=\"$url/image/folder.gif\">
    </td><td width=100 align=left>$file</td>\n";
    print "<td width=50>",$size,"</td><td width=200>",$atime,"</td>
    <td width=200>",$mtime,"</td></tr>\n";
    }
    }
    print "</table>\n";
# Ту ж операцію проводимо для файлів. Друкуємо шапку таблиці.
    print <<HTML;
    <p><table bgcolor=\"bfbfbf\" width=600 border
    cellspacing=0 cellpadding=0>
    <tr><td colspan=5 align=center><b><font color="ff0000">Files</font>
    </b></td></tr>
    <tr><td>.</td><td><b>List</b><td><b>Size</b>
    <td><b>Last accessed</b><td><b>Last modified</b></tr>
#HTML
    foreach $file(@files) {
    $size=(stat("$userdir/$file"))[7];
    $atime=localtime((stat("$userdir/$file"))[8]);
    $mtime=localtime((stat("$userdir/$file"))[9]);
    if (!-d "$userdir/$file" && "$file" ne "." && "$file" ne "..") {
# поміщаємо знайдені файли в масив
        push (@dir,"$userdir/$file");
# підраховуємо їх кількість.
        $number=@dir;
# Видаємо інформацію.
        print "<tr><td width=30><img src=\"$url/image/page.gif\"></td>
        <td width=100><a href=\"$basedir$login/$file\">",$file,"</a></td>\n";
        print "<td width=50>",$size,"</td>\n";
        print "<td width=200>",$atime,"</td>
        <td width=200>",$mtime,"</td></tr>\n";
        }
        }
    print "</table>\n";
    print "<p><center><b><font color=\"0000ff\">There are",
    $number," files in this directory.</b></font></center>\n";
    }

```

Алгоритми були створені для системи моніторингу трафіку локальної мережі. Читання й запис файлів і каталогів є чи не головною перевагою Perl. Практично будь-який Perl – скрипт використовує або запис у файли, або пошук певних рядків у файлі, або запис або завантаження файлів у каталоги.

Файл можна відкривати для читання, запису, додавання або пошуку певних рядків:

1. Відкрити файл для читання – `open (FILE,"$file");`
2. Відкрити файл для запису – `open (FILE,">$file");`
3. Відкрити файл для додавання – `open (FILE,">>$file");`
4. Відкрити файл для читання й додавання – `open (FILE,"+<$file").`

Де:

– FILE це дескриптор або описатель файлу, тобто ім'я, під яким він фігурує в програмі. Може бути будь-яким рядком символів, рекомендується набирати прописними буквами щоб уникнути неоднозначностей;

– \$file змінна для файлу, містить звичайне ім'я файлу й шлях до нього. Переважно оголосити цю змінну на початку скрипта. Це зручно, якщо файл перебуває десь глибоко в підкаталогах, приміром, незручно весь час набирати `/usr/local/htdocs/main/pages/file.html`. Зміст файлу можна читати порядково або в масив:

```
#!/usr/local/bin/perl
# прочитати один рядок з файлу.
open (STAT,"$file");
$count=;
close (STAT);
```

```
-----
#!/usr/local/bin/perl
# прочитати файл у масив.
open (STAT,"$file");
@count=;
close (STAT);
```

Розглянемо сценарій реєстрації користувача на веб-сервері. Ім'я користувача і його пароль записуються в текстовий файл і використовуються для його наступної автентифікації.

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

```

#!/usr/local/bin/perl
# об'ява глобальних змінних
$request=$ENV{'REQUEST_METHOD'};
$content=$ENV{'CONTENT_LENGTH'};
$basedir="http://www.mydomain.com/~";
$userdir="f:/home";

# Підпрограми для декодування даних з форми.
sub urldecode {
    local($val)=@_;
    $val=~ s/\+/ /g;
    $val=~ s/%[0-9a-ha-h] {2}/pack('C',hex($1))/ge;
    return $val;
}

sub strhtml {
    local($val)=@_;
    $val=~s//>/g;
    $val=~s/(http:\\/\\/\\+S)/<A href="$1">$1</A>/g;
    return $val;
}

if ($request eq 'GET') {
    $query=$ENV{'QUERY_STRING'};
}
else {
    sysread(STDIN,$query,$content);
}

# Генеруємо форму, якщо ніякі дані не введені.
print "Content-type:text/html\n\n";
print <<HTML_gen;
<HTML><BODY bgcolor="e6e8fa">
HTML_gen
if ($query eq '') {
print <<HTML;
<h2 align=center><font color="ff0000">Registration.</font></h2>
<p><font face="serif" size=2> Please,fill in the form below.
<p>After registration you will receive your personal directory</font>
<p><FORM ACTION=" ../cgi-bin/addlogin.cgi" METHOD="POST" name="reg">
<center><TABLE BGCOLOR="bfbfbf">
<TR><td><font color="ff0000">*</font>
<TD><b>Login:</b><TD><INPUT TYPE="text" NAME="login" SIZE="20">
<TR><td><font color="ff0000">*</font>
<TD><b>Password:</b>

```

						ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			57

```

<TD><INPUT TYPE="password" NAME="pass" SIZE="20">
<TR><td><font color="ff0000">*</font><TD><b>E-mail:</b>
<TD><INPUT TYPE="text" NAME="email" SIZE="20">
<TR><TD colspan=3><p><center>
<INPUT TYPE="submit" VALUE="Submit"></center>
</TABLE></center>
</FORM>
# HTML
# Декодуємо поля форми
    else {
        foreach (@fields=split(/&/,$query)) {
            if (/login=(.*)/) { $login=&urldecode ($1); }
            if (/pass=(.*)/) { $password=&urldecode ($1); }
            if (/email=(.*)/) { $email=&urldecode ($1); }
        }
# Перевіряємо, не чи існує дане ім'я в системі.
        open(INFO,"login.txt") ||die;
# читаємо рядок в масив.
        @data=<INFO>;
        close(INFO);
        foreach $string(@data) {
            @item=split(/&/,$string);
# Розбиваємо рядок на частини
            foreach (@item) {
                if ($item[0] eq $login) {
# Порівнюємо отримане ім'я з першим полем файлу
# для кожного рядка і якщо таке знайдене видаємо
# помилку.
                print <<HTML;
                <h2 align=center><font color="ff0000">Error!</font></h2>
                <p><center><b>The name <font color="ff0000">$login</font> already
                exists in the system. <p>Please,go back and choose another name.</b>
                <p><form><input type="button" value="Back" onclick=
                "history.back()"></form>
                </center>
                HTML
                exit;
                }
                }
                }
# Якщо ім'я не знайдене,відкриваємо базу даних і додаємо інформацію.
                if ($item[0] ne $login) {

```

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

```

open(DATA,">>login.txt");
$string=join('&', $login, $password, $email, scalar localtime, $ENV
           {'REMOTE_ADDR'});
print DATA "$string\n";
close(DATA);
# Створюємо домашній каталог користувача й переходимо в нього.
mkdir("$userdir/$login",0700);
chdir("$userdir/login");
opendir(USER,"$userdir/$login");
# Поміщаємо файл index.html у каталог користувача.
open(IN,">$userdir/$login/index.html");
print IN "This is the test!\n";
close(IN);
closedir(USER);
# Зміст файлу може бути будь-яким,це тільки для прикладу.
# Генеруємо відповідь користувачеві.
print <<HTML;
<p><h1 align=center><font color="ff0000">Congratulations!</font></h1>
<p><b>Your registration was successful and your data were added to
our database.Thank you for your time.</b>
<p><center><b><font color="ff0000">
You entered:</font>(print this page and keep it in safe place)</b>
<p><table>
<tr><td><b>Your login name:</b><td><font color="0000ff">$login</font>
<tr><td><b>Your password:</b><td><font color="0000ff">$password</font>
<tr><td><b>Your e-mail address:</b>
<td><font color="0000ff">$email</font>
</table></center>
HTML
}
}

```

Скрипт видає відповідь у вигляді html сторінки, що містить всю інформацію, введену користувачем.

4.2 Захист розробленого програмного забезпечення

Дані в програмі захищаються за допомогою використання алгоритму CAST-128 (або CAST5) у криптографії, це блоковий алгоритм симетричного шифрування на основі мережі Фейстеля, який використовується в цілому ряді

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

продуктів криптографічного захисту, зокрема деяких версіях PGP і GPG і крім того схвалений для використання Канадським урядом.

Основні відомості

Алгоритм був створений в 1996 році Карлайлом Адамсом (Carlisle Adams) і Стаффордом Таваресом (Stafford Tavares) використовуючи метод побудови шифрів CAST, який використовується також і іншим їхнім алгоритмом CAST-256 (алгоритм-кандидат AES).

CAST-128 складається з 12 або 16 раундів мережі Фейстеля з розміром блоку 64 біта й довжиною ключа від 40 до 128 біт (але тільки з інкрементацією по 8 біт). 16 раундів використовуються коли розміри ключа перевищують 80 біт. В алгоритмі використовуються 8x16 S- блоки, засновані на бент-функції, операції XOR і модулярної арифметиці (модулярне додавання й вирахування). Є три різні типи функцій раундів, але вони схожі за структурою й різняться тільки у виборі виконуваної операції (додавання, вирахування або XOR) у різних місцях.

Хоча CAST-128 захищений патентом Entrust, його можна використовувати в усьому світі для комерційних або некомерційних цілей безкоштовно.

Опис

CAST – це популярний 64-бітовий шифр, що допускає розміри ключа аж до 128 біт

Алгоритм CAST використовує 64-бітовий блок і 64-бітовий ключ. CAST стійкий до диференціального й лінійного криптоаналізу. Сила алгоритму CAST укладена в його S-блоках. В CAST немає фіксованих S-блоків і для кожного додатка вони конструюються заново. Створений для конкретної реалізації CAST S-блок уже більше ніколи не міняється. Інакше кажучи, S-блоки залежать від реалізації, а не від ключа. Northern Telecom використовує CAST у своєму пакеті програм Entrust для комп'ютерів Macintosh, PC і робочих станцій UNIX. Обрані ними S-блоки не опубліковані, що втім не дивно.

CAST-128 належить компанії Entrust Technologies, але є безкоштовним як для комерційного, так і для некомерційного використання. CAST-256 –

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

Раунди	$I = ((Km_i + R_{i-1}) \lll Kr_i)$
1,4,7,10,13,16	$F = ((S1[I_a] \ S2[I_b]) - (S3[I_c])) + S4[I_d]$
Раунди	$I = ((Km_i \wedge R_{i-1}) \lll Kr_i)$
2,5,8,11,14	$F = ((S1[I_a] - S2[I_b]) + (S3[I_c])) \ S4[I_d]$
Раунди	$I = ((Km_i - R_{i-1}) \lll Kr_i)$
3,6,9,12,15	$F = ((S1[I_a] + S2[I_b]) \ (S3[I_c])) - S4[I_d]$

Поля заміни

CAST-128 використовує вісім полів заміни: поля S1, S2, S3 і S4 раундові функції полів заміни, S5, S6, S7 і S8 є ключами розгорнення полів заміни. Незважаючи на те, що 8 полів заміни вимагають у цілому 8 Кбайт для зберігання, зверніть увагу на те, що тільки 4 Кбайта потрібні під час фактичного шифрування / дешифрування, тому що генерація підключа звичайно робиться до будь-якого введення даних.

Ключі розгорнення

Представимо 128-розрядний ключ у вигляді $x_0x_1x_2x_3x_4x_5x_6x_7x_8x_9x_{10}x_{11}x_{12}x_{13}x_{14}x_{15}x_{16}x_{17}$, де x_0 старший байт, і x_{17} молодший байт.

Представимо $z_0..z_{17}$ проміжними (тимчасовими) байтами. $S_i[]$ представляє поле заміни і \wedge представляє додавання по Хор'у.

Поля заміни формуються із ключа $x_0x_1x_2x_3x_4x_5x_6x_7x_8x_9x_{10}x_{11}x_{12}x_{13}x_{14}x_{15}x_{16}x_{17}$ у такий спосіб.

$$z_0z_1z_2z_3 = x_0x_1x_2x_3 \wedge S5[xD] \wedge S6[xF] \wedge S7[xC] \wedge S8[xE] \wedge S7[x8]$$

$$z_4z_5z_6z_7 = x_8x_9x_{10}x_{11} \wedge S5[z_0] \wedge S6[z_2] \wedge S7[z_1] \wedge S8[z_3] \wedge S8[xA]$$

$$z_8z_9z_{10}z_{11} = x_{12}x_{13}x_{14}x_{15} \wedge S5[z_7] \wedge S6[z_6] \wedge S7[z_5] \wedge S8[z_4] \wedge S5[x_9]$$

$$z_{12}z_{13}z_{14}z_{15} = x_{16}x_{17}x_{18}x_{19} \wedge S5[zA] \wedge S6[z_9] \wedge S7[zB] \wedge S8[z_8] \wedge S6[xB]$$

$$K1 = S5[z_8] \wedge S6[z_9] \wedge S7[z_7] \wedge S8[z_6] \wedge S5[z_2]$$

$$K2 = S5[zA] \wedge S6[zB] \wedge S7[z_5] \wedge S8[z_4] \wedge S6[z_6]$$

$$K3 = S5[zC] \wedge S6[zD] \wedge S7[z_3] \wedge S8[z_2] \wedge S7[z_9]$$

$$K4 = S5[zE] \wedge S6[zF] \wedge S7[z_1] \wedge S8[z_0] \wedge S8[zC]$$

$$x_0x_1x_2x_3 = z_8z_9z_{10}z_{11} \wedge S5[z_5] \wedge S6[z_7] \wedge S7[z_4] \wedge S8[z_6] \wedge S7[z_0]$$

$$\begin{aligned}
x_4x_5x_6x_7 &= z_0z_1z_2z_3 \wedge S_5[x_0] \wedge S_6[x_2] \wedge S_7[x_1] \wedge S_8[x_3] \wedge S_8[z_2] \\
x_8x_9xAxB &= z_4z_5z_6z_7 \wedge S_5[x_7] \wedge S_6[x_6] \wedge S_7[x_5] \wedge S_8[x_4] \wedge S_5[z_1] \\
xCxDxEzF &= zCzDzEzF \wedge S_5[xA] \wedge S_6[x_9] \wedge S_7[xB] \wedge S_8[x_8] \wedge S_6[z_3] \\
K_5 &= S_5[x_3] \wedge S_6[x_2] \wedge S_7[xC] \wedge S_8[xD] \wedge S_5[x_8] \\
K_6 &= S_5[x_1] \wedge S_6[x_0] \wedge S_7[xE] \wedge S_8[xF] \wedge S_6[xD] \\
K_7 &= S_5[x_7] \wedge S_6[x_6] \wedge S_7[x_8] \wedge S_8[x_9] \wedge S_7[x_3] \\
K_8 &= S_5[x_5] \wedge S_6[x_4] \wedge S_7[xA] \wedge S_8[xB] \wedge S_8[x_7] \\
z_0z_1z_2z_3 &= x_0x_1x_2x_3 \wedge S_5[xD] \wedge S_6[xF] \wedge S_7[xC] \wedge S_8[xE] \wedge S_7[x_8] \\
z_4z_5z_6z_7 &= x_8x_9xAxB \wedge S_5[z_0] \wedge S_6[z_2] \wedge S_7[z_1] \wedge S_8[z_3] \wedge S_8[xA] \\
z_8z_9AzB &= xCxDxEzF \wedge S_5[z_7] \wedge S_6[z_6] \wedge S_7[z_5] \wedge S_8[z_4] \wedge S_5[x_9] \\
zCzDzEzF &= x_4x_5x_6x_7 \wedge S_5[zA] \wedge S_6[z_9] \wedge S_7[zB] \wedge S_8[z_8] \wedge S_6[xB] \\
K_9 &= S_5[z_3] \wedge S_6[z_2] \wedge S_7[zC] \wedge S_8[zD] \wedge S_5[z_9] \\
K_{10} &= S_5[z_1] \wedge S_6[z_0] \wedge S_7[zE] \wedge S_8[zF] \wedge S_6[zC] \\
K_{11} &= S_5[z_7] \wedge S_6[z_6] \wedge S_7[z_8] \wedge S_8[z_9] \wedge S_7[z_2] \\
K_{12} &= S_5[z_5] \wedge S_6[z_4] \wedge S_7[zA] \wedge S_8[zB] \wedge S_8[z_6] \\
x_0x_1x_2x_3 &= z_8z_9AzB \wedge S_5[z_5] \wedge S_6[z_7] \wedge S_7[z_4] \wedge S_8[z_6] \wedge S_7[z_0] \\
x_4x_5x_6x_7 &= z_0z_1z_2z_3 \wedge S_5[x_0] \wedge S_6[x_2] \wedge S_7[x_1] \wedge S_8[x_3] \wedge S_8[z_2] \\
x_8x_9xAxB &= z_4z_5z_6z_7 \wedge S_5[x_7] \wedge S_6[x_6] \wedge S_7[x_5] \wedge S_8[x_4] \wedge S_5[z_1] \\
xCxDxEzF &= zCzDzEzF \wedge S_5[xA] \wedge S_6[x_9] \wedge S_7[xB] \wedge S_8[x_8] \wedge S_6[z_3] \\
K_{13} &= S_5[x_8] \wedge S_6[x_9] \wedge S_7[x_7] \wedge S_8[x_6] \wedge S_5[x_3] \\
K_{14} &= S_5[xA] \wedge S_6[xB] \wedge S_7[x_5] \wedge S_8[x_4] \wedge S_6[x_7] \\
K_{15} &= S_5[xC] \wedge S_6[xD] \wedge S_7[x_3] \wedge S_8[x_2] \wedge S_7[x_8] \\
K_{16} &= S_5[xE] \wedge S_6[xF] \wedge S_7[x_1] \wedge S_8[x_0] \wedge S_8[xD]
\end{aligned}$$

половина, що залишається, ідентична тому, що дане вище, продовження від останнього створило $x_0..x_f$, щоб генерувати ключі $K_{17} - K_{32}$.

$$\begin{aligned}
z_0z_1z_2z_3 &= x_0x_1x_2x_3 \wedge S_5[xD] \wedge S_6[xF] \wedge S_7[xC] \wedge S_8[xE] \wedge S_7[x_8] \\
z_4z_5z_6z_7 &= x_8x_9xAxB \wedge S_5[z_0] \wedge S_6[z_2] \wedge S_7[z_1] \wedge S_8[z_3] \wedge S_8[xA] \\
z_8z_9AzB &= xCxDxEzF \wedge S_5[z_7] \wedge S_6[z_6] \wedge S_7[z_5] \wedge S_8[z_4] \wedge S_5[x_9] \\
zCzDzEzF &= x_4x_5x_6x_7 \wedge S_5[zA] \wedge S_6[z_9] \wedge S_7[zB] \wedge S_8[z_8] \wedge S_6[xB]
\end{aligned}$$

$K17 = S5[z8] \wedge S6[z9] \wedge S7[z7] \wedge S8[z6] \wedge S5[z2]$
 $K18 = S5[zA] \wedge S6[zB] \wedge S7[z5] \wedge S8[z4] \wedge S6[z6]$
 $K19 = S5[zC] \wedge S6[zD] \wedge S7[z3] \wedge S8[z2] \wedge S7[z9]$
 $K20 = S5[zE] \wedge S6[zF] \wedge S7[z1] \wedge S8[z0] \wedge S8[zC]$
 $x0x1x2x3 = z8z9zAzB \wedge S5[z5] \wedge S6[z7] \wedge S7[z4] \wedge S8[z6] \wedge S7[z0]$
 $x4x5x6x7 = z0z1z2z3 \wedge S5[x0] \wedge S6[x2] \wedge S7[x1] \wedge S8[x3] \wedge S8[z2]$
 $x8x9xAxB = z4z5z6z7 \wedge S5[x7] \wedge S6[x6] \wedge S7[x5] \wedge S8[x4] \wedge S5[z1]$
 $xCxDxEzF = zCzDzEzF \wedge S5[xA] \wedge S6[x9] \wedge S7[xB] \wedge S8[x8] \wedge S6[z3]$
 $K21 = S5[x3] \wedge S6[x2] \wedge S7[xC] \wedge S8[xD] \wedge S5[x8]$
 $K22 = S5[x1] \wedge S6[x0] \wedge S7[xE] \wedge S8[xF] \wedge S6[xD]$
 $K23 = S5[x7] \wedge S6[x6] \wedge S7[x8] \wedge S8[x9] \wedge S7[x3]$
 $K24 = S5[x5] \wedge S6[x4] \wedge S7[xA] \wedge S8[xB] \wedge S8[x7]$
 $z0z1z2z3 = x0x1x2x3 \wedge S5[xD] \wedge S6[xF] \wedge S7[xC] \wedge S8[xE] \wedge S7[x8]$
 $z4z5z6z7 = x8x9xAxB \wedge S5[z0] \wedge S6[z2] \wedge S7[z1] \wedge S8[z3] \wedge S8[xA]$
 $z8z9zAzB = xCxDxEzF \wedge S5[z7] \wedge S6[z6] \wedge S7[z5] \wedge S8[z4] \wedge S5[x9]$
 $zCzDzEzF = x4x5x6x7 \wedge S5[zA] \wedge S6[z9] \wedge S7[zB] \wedge S8[z8] \wedge S6[xB]$
 $K25 = S5[z3] \wedge S6[z2] \wedge S7[zC] \wedge S8[zD] \wedge S5[z9]$
 $K26 = S5[z1] \wedge S6[z0] \wedge S7[zE] \wedge S8[zF] \wedge S6[zC]$
 $K27 = S5[z7] \wedge S6[z6] \wedge S7[z8] \wedge S8[z9] \wedge S7[z2]$
 $K28 = S5[z5] \wedge S6[z4] \wedge S7[zA] \wedge S8[zB] \wedge S8[z6]$
 $x0x1x2x3 = z8z9zAzB \wedge S5[z5] \wedge S6[z7] \wedge S7[z4] \wedge S8[z6] \wedge S7[z0]$
 $x4x5x6x7 = z0z1z2z3 \wedge S5[x0] \wedge S6[x2] \wedge S7[x1] \wedge S8[x3] \wedge S8[z2]$
 $x8x9xAxB = z4z5z6z7 \wedge S5[x7] \wedge S6[x6] \wedge S7[x5] \wedge S8[x4] \wedge S5[z1]$
 $xCxDxEzF = zCzDzEzF \wedge S5[xA] \wedge S6[x9] \wedge S7[xB] \wedge S8[x8] \wedge S6[z3]$
 $K29 = S5[x8] \wedge S6[x9] \wedge S7[x7] \wedge S8[x6] \wedge S5[x3]$
 $K30 = S5[xA] \wedge S6[xB] \wedge S7[x5] \wedge S8[x4] \wedge S6[x7]$
 $K31 = S5[xC] \wedge S6[xD] \wedge S7[x3] \wedge S8[x2] \wedge S7[x8]$
 $K32 = S5[xE] \wedge S6[xF] \wedge S7[x1] \wedge S8[x0] \wedge S8[xD]$

Маскування й перестановка підключів

$K_{m_1}, \dots, K_{m_{16}}$ 32-розрядні підключи маскування (один на раунд).

$K_{r_1}, \dots, K_{r_{16}}$ 32-розрядні перестановки підключів (один на раунд); тільки молодші 5 бітів використовуються в кожному раунді.

```
for (i=1; i<=16; i++) {  $K_{m_i} = K_i$ ;  $K_{r_i} = K_{16+i}$ ; }
```

Змінний розмір ключа

CAST-128 Алгоритм шифрування був розроблений, щоб розмір ключа міг варіюватися від 40 до 128 біт, в 8-бітному кроці (тобто припустимі розміри ключа рівняються 40, 48, 56, 64..., 112, 120, і 128 бітам). Для змінної роботи розміру ключа специфікація наступні:

- 1) Для розмірів ключа до й включаючи 80 бітів (тобто, 40, 48, 56, 64, 72, і 80 бітів) алгоритм точно такої ж, але використовує 12 раундів замість 16;
- 2) Для розмірів ключа більше, чим 80 бітів, алгоритм використовує повні 16 раундів;
- 3) Для розмірів ключа менше, чим 128 бітів ключ доповнений нульовими байтами (у самих правих, або молодших, позиціях) до 128 біток (тому що розклад ключа CAST 128 ухвалює вхідний ключ 128 бітів).

Розшифрування

Розшифрування для CAST-128 відносно проста. Розшифрування працює в тому ж алгоритмічному напрямку, що й шифрування, починаючи із зашифрованого тексту як вхідних даних. При цьому підключ використовуються у зворотному напрямку.

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено головне вікно програми. З нього видно, що інтерфейс користувача програми складається з таких логічних блоків:

- Меню програми яке складається з розділів: Файл; ЛМ (локальна мережа); Аналіз; Довідка.
- Функціональних клавiш, а саме: Підключитись; Відключитись; Статистика доступу; Налаштування; Журнал роботи програми.
- Обрання основної мережної карти.
- Поточної інформації.
- Кнопки сканування – отримання поточних даних.

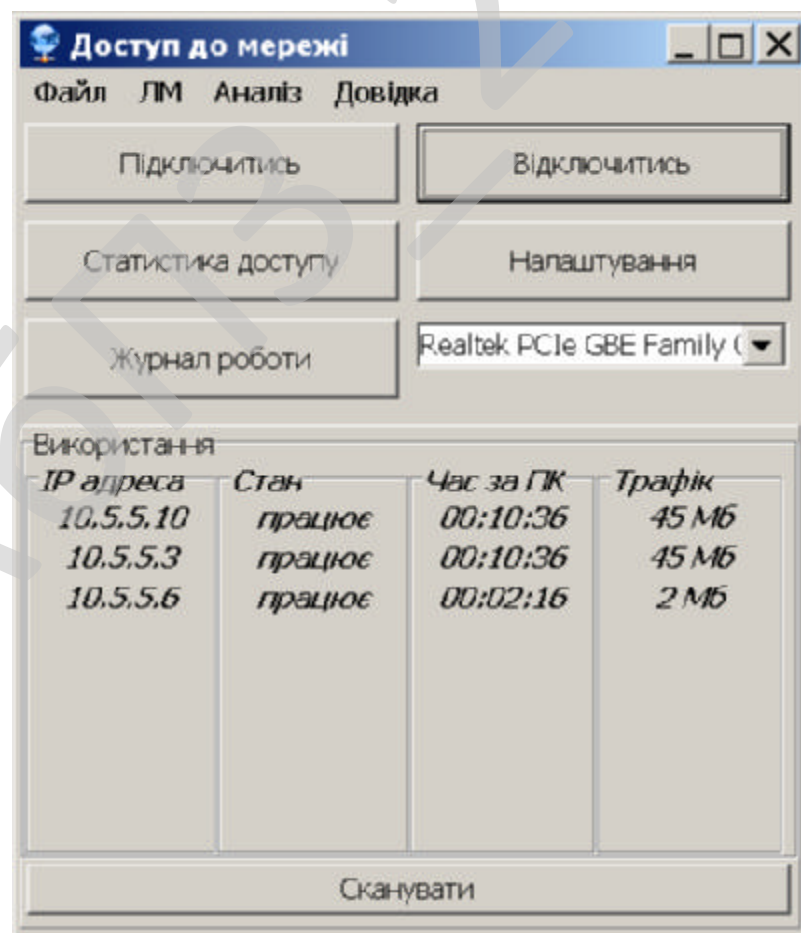


Рисунок 5.1 – Головне вікно програми

Перевагою розробленого ПЗ є те, що воно використовує безпечний протокол РРТР. Протокол РРТР дозволяє створювати захищені канали для обміну даними по різних мережевих протоколах – IP, IPX або NetBEUI.

Дані цих протоколів інкапсулюються за допомогою протоколу РРТР в пакети протоколу IP, за допомогою якого переносяться в зашифрованому вигляді через мережу TCP / IP. Інкапсулюється вихідний кадр PPP, тому протокол РРТР можна віднести до класу протоколів інкапсуляції канального рівня в мережевий.

На рисунку 5.2 зображено форму авторського права.

Обрано тип розповсюдження – безплатне програмне забезпечення (Freeware). Це власницьке програмне забезпечення, котре можна безоплатно використовувати протягом необмеженого терміну без обмежень у функціональності, і поширюване без вихідних кодів. Автори такого програмного забезпечення, як правило, хочуть «дати щось спільноті», але хочуть також контролювати його подальшу розробку. Іноді, коли програмісти вирішують припинити розробку, вони передають сирцевий код іншим програмістам, або ж спільноті як вільне програмне забезпечення.

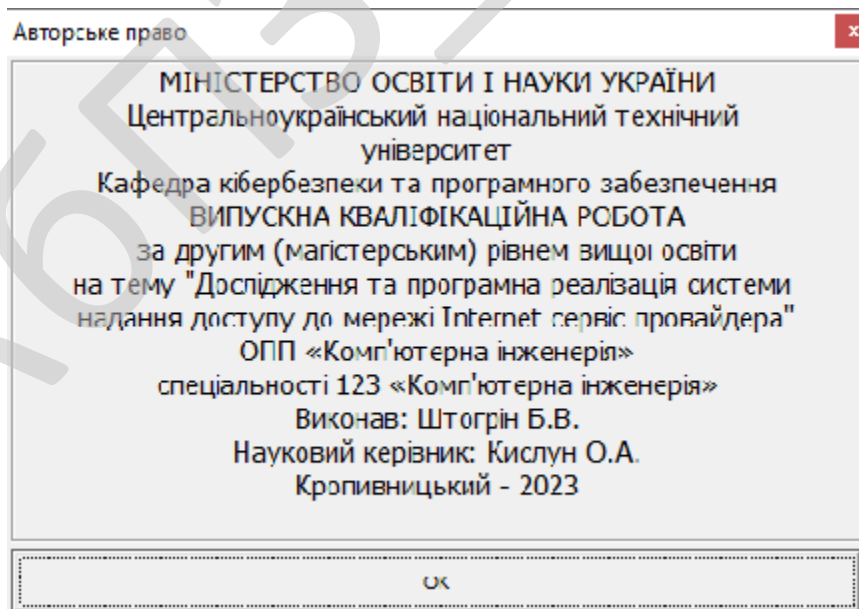


Рисунок 5.2 – Довідка програми

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи надання доступу до мережі Internet сервіс провайдера.

Метою розробки є дослідження та програмна реалізація системи надання доступу до мережі Internet сервіс провайдера.

Об'єктом дослідження є процес надання доступу до мережі Internet сервіс провайдера.

Предметом дослідження є методи надання доступу до мережі Internet сервіс провайдера.

Методи дослідження базуються на методах теорії комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.

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

– Удосконалено метод надання доступу до мережі Internet сервіс провайдера.

– Розроблено вітчизняний продукт надання доступу до мережі Internet сервіс провайдера, який має більш широкі можливості, на відміну від існуючих аналогів.

					VKPM-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

Продовження таблиці 7.1

1	2	3
7. Кількість макетів вхідної інформації	–	3
8. Кількість форм вихідної інформації.	–	4
9. Мова програмування (1-6)	–	1
10. Попередній досвід (1-6)	–	3
11. Гнучкість проекту ПП (1-6)	–	3
12. Детальність проекту ПП (1-6)	–	2
13. Рівень спрацьованості колективу (1-6)	–	2
14. Ступінь вимірності процесів (1-6)	–	3
15. Необхідна надійність програмного забезпечення (1-6)	–	2
16. Розмір бази даних (порівняно з розміром програми) (1-6)	–	2
17. Складність кінцевого програмного продукту (1-6)	–	2
18. Необхідний рівень забезпечення повторного використання (1-6)	–	2
19. Документованість відповідно до планованого життєвого циклу (1-6)	–	2
20. Вимоги до швидкодії ПП (1-6)	–	2
21. Обмеження на розміри основного сховища даних (1-6)	–	2
22. Різноманітність використовуваних обчислювальних платформ (1-6)	–	2
23. Професійний рівень аналітиків (1-6)	–	2
24. Професійний рівень програмістів (1-6)	–	2
25. Постійність складу команди розробників (1-6)	–	2
26. Досвід розробки додатків (1-6)	–	2
27. Досвід роботи з обчислювальною платформою (1-6)	–	2

Продовження таблиці 7.1

1	2	3
28. Досвід роботи з мовою і інструментами середовища розробки (1-6)	–	2
29. Досвід роботи з програмними інструментами розробки (1-6)	–	3
30. Розробка ПЗ для декількох серверів одночасно (1-6)	–	2
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн.	–	19000
33. Норматив додаткової зарплати, % :	Н _д	10
34. Норматив відрахувань у соціальні фонди, %	Н _с	22
35. Норматив загальногосподарських витрат, %	Н _г	15
36. Норматив витрат на освоєння нових мов програмування, %	Н _п	15
37. Рівень рентабельності програмної продукції, %	Р _е	50
38. Ставка податку на додану вартість, %	Н _{дв}	20

7.2 Розрахунок трудомісткості розробки програмної продукції

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

Визначимо трудомісткість розробки ПЗ для стадії РП.

Обчислюємо номінальні трудовитрати, люд-міс.:

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

де: A – коефіцієнт Боєма, $A = 2,45$;

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

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:

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

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

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

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

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

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

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

Найменування обладнання	Профілактичне обслуговування			
	Кількість хв. на один. обл.	Кількість обладнання	Затрати часу в хв.	Затрати часу в год.
Системний блок ПК	90	6	540	9
Монітор	60	6	360	6
Клавіатура	30	6	180	3
Маніпулятор «мишка»	30	6	180	3
Принтер матричний	60	0	0	0,0
Принтер лазерний	120	3	360	6
Принтер струминний	60	1	60	1
Сканер	20	1	20	0,33
Концентратор-маршрутизатор	30	4	120	2
Кабельні господарства ЛОМ на 1 м. п.	2,5	550	1375	22,92
Копіювальний апарат	140	2	280	4,67
Усього за рік:			3 _ч	57,92

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:

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

$$\Phi_{op}^c = \frac{58 \cdot 3}{1,2} = 145 \text{ год.}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

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

$$Ч_{ел} = 145 / (60 \cdot 8) = 0,3 \text{ ставки.}$$

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 2022, серверу доступу ADSL (OC Linux), налаштування ADSL, VPN PPPoE, Frame Relay, Wi-Fi	2	0,5
	Налаштування і конфігурування базової станції безпроводного зв'язку (CMTS)	0,5	
	Розробка та впровадження проектів з організації зв'язку між віддаленими об'єктами, ЛОМ	0,5	
	Забезпечення цілодобової роботи зв'язку клієнтів до мережі Інтернет	1	
Всього		4	

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньомісячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	1	12475	37425
Продакт-менеджер	0,25	12000	9000
Інженер-програміст	3,8	12000	136800
Інженер-електронщик	0,3	11500	10350
Інженер-системотехнік	0,25	11500	8625
Адміністратор мережі	0,5	11500	17250
Системний програміст	0,25	11500	8625
Дизайнер WEB	0,25	12000	9000
Інженер-верстальник	0,25	11700	8775
Бухгалтер-економіст	0,5	12500	18750
Всього за період розробки	$R_{cn} = 7,35$	-	$\Phi_{роб} = 264600$

Розрахуємо середньоденну зарплату одного виконавця:

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

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

$$z_{cd} = \frac{264600}{7,35 \cdot 60} = 600 \text{ грн.}$$

7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника

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

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

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

де: R_{cn}^1 – кількість робочих місць виконавців, шт. Приймаємо 8 робочих місць;
 S_y – питома площа на одне робоче місце, m^2 ;
 $C_{пл}$ – вартість одного квадратного метра площі, грн.

Згідно даних ТОВ науково-дослідницького консалтингового підприємства «Пектораль» (м. Кіровоград, вул. Глинки 16) ціна одного квадратного метра площі новобудови, вік якої не перевищує 25 років, по місту складає 800...1600 $у.о./m^2$. Враховуючи, що курс складає 1 $у.о.$ = 25 грн. приймаємо для розрахунку вартість одного метра квадратного рівною 20000 грн./ m^2 . На кожне робоче місце у середньому потрібно $8 m^2$. З урахуванням цього:

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

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

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

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

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

$$I_{нв} = 8 \cdot 12500 = 100000 \text{ грн.}$$

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

Специфікація на обчислювальну техніку наведена в таблиці 7.7.

Дані по оптовій ціні на обладнання та комплектуючі вибирались по прайсу фірми Комп'ютерторг за 15.10.23 – джерело <http://computorg.ua/price.html>

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Персональний комп'ютер		10947
Системний блок		7347
Процесор	Intel Celeron G5905 (BX80701G5905) 1200, 2 ядра, 2 потоки, 3.5 GHz, TDP – 58 Вт, 14nm box	-
Системна плата	GIGABYTE H470M H 1200, DDR4 64 ГБ, 3200 MHz, LAN – 1 Гбит/с, D-Sub (VGA), HDMI, 1 x M.2 2280, 4 x Sata 6.0 Gb/s, Micro-ATX	-
Відеокарта	Radeon R7 350 2Gb Afox (AFR7350-2048D5H4-V3) PCI-Express 3.0, 2 ГБ, GDDR5, 128 Bit	-
Жорсткий диск	SSD 2.5" 256GB Mibrand, 3D TLC NAND, 2.5", SATA III (6Gb/s)	-
Оперативна пам'ять	DDR4 8GB 2666 MHZ GOODRAM (GR2666D464L19S/8G)	-
DVD-привод	-	-
Корпус	ATX Middle Tower FOXCONN Pro, 3GTLA-489, PSU 350W(FSP Brand: ATX-350PNR, 12cm), black, (front bezel – black+light silver; body material – 0.6mm), 80mm fan (rear), 2xUSB2.0/AUDIO/MIC, Air Duct, Tool-less chassis design, Thermally Advantaged Chassis	-

Продовження таблиці 7.6

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Кулер	–	–
Кардрідер внутрішній	USB 2.0 Card reader STORM CR-35U1A4-B, int. 3.5", 1*USB2.0+AUDIO+1394, multi: All Type Cards, black	220
інше	Клавіатура, мишка	Подарунок
Монітор	22" TFT, ASUS VW223D (5ms, 300/3000:1, 170/160, D-SUB, Wide)	3600
Принтер лазерний	Canon i-SENSYS LBP6030W	2700
Принтер струминний	Epson Stylus Photo P50 (C11CA45341) + USB cable	5500
Копіювальний апарат	Canon i-SENSYS MF217W with Wi-Fi	5965

Таблиця 7.7 – Балансова вартість обчислювальної техніки

Найменування обчислювальної техніки	Кількість, шт.	Ціна за одиницю, грн.	Витрати на транспортування, монтаж та випробовування.	Загальна вартість, грн.
Персональні комп'ютери	8	10947	8757,6	96333,6
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Сканери	-	-	-	0
Копіюв. апарат	1	5965	596,5	6561,5
Всього	–	–	–	114885,1

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 7.8 – 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
Група 4			
3. Обчислювальна техніка	114885	-	-
Всього по групі	114885	50	57442,5
Група 5, 6			
4. Вимірювальні пристрої	5190	25	1297,5
5. Транспортні засоби	97500	20	19500
6. Господарський інвентар	100000	25	25000
Всього по групі	202690	-	45797,5
7. Нематеріальні активи	19000	10	1900
Разом	$K_p = 1744575$		$A_p = 175540$

Примітка: вартість автомобіля взята по даним з автосалону автотрейдинг, вкладки автобазар, джерело <http://www.auto-trading.com.ua/sale/lot20772.html>, складає 2635 USD, що враховуючи курс 37 складає 97500 грн.

7.5 Визначення собівартості розробки та ціни програмної продукції

Визначимо основну зарплату виконавців:

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

де: N_e – кількість екземплярів програм, шт.

$$Z_o = 600 \cdot 209 / 19 = 6600 \text{ грн.}$$

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, %.

$$Z_d = 6600 \cdot 10 \cdot 0,01 = 660 \text{ грн.}$$

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, %.

$$C_{oc} = 0,01 \cdot 22(6600 + 660) = 2686 \text{ грн.}$$

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, %.

$$G_{ocn} = 6600 \cdot 15 \cdot 0,01 = 990 \text{ грн.}$$

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

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

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

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 норм приймаємо 0,5 пачки паперу на три місяці розробки. Тоді, враховуючи, що вартість пачки паперу складає $Ц_n = 210$ грн., визначаємо вартість паперу за період розробки $N_m = 3$ міс:

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

$$Z_{M1} = 210 \cdot 0,5 = 105 \text{ грн.}$$

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

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

де: $Ц_\delta$ – вартість дисків CD/DVD: CDR TDK 700Mb, 80Min, 52x Cake box – 28 грн./шт., DVD-R LG 4,7Gb, 16x speed Cake box – 51 грн./шт.

$$Z_{M2} = 28 \cdot 2 + 51 = 107 \text{ грн.}$$

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

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

де: $Ц_\epsilon$ – 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.

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

$$Z_M = (105 + 107 + 1702)/19 = 101 \text{ грн.}$$

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, %.

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83

$$O_n = 6600 \cdot 15 \cdot 0,01 = 990 \text{ грн.}$$

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	6600
2. Додаткова зарплата виконавців	Z_d	660
3. Відрахування на соціальні потреби	C_{oc}	2686
4. Загальногосподарські витрати	Γ_{ocn}	990
5. Витрати на матеріали	Z_M	101
6. Освоєння нових операційних систем, мов програмування	O_n	990
7. Амортизація основних фондів	A_m	2310
8. Повна собівартість програмного забезпечення	C_n	14337
9. Плановий прибуток	Π_p	7168,5
10. Ціна підприємства $C_n = C_n + \Pi_p$	C_n	21505,5
11. Податок на додану вартість $\Pi Д В = 0,01 \cdot H_{oc} \cdot C_n$	$\Pi Д В$	4301,1
12. Відпускна ціна програмної продукції $C = C_n + \Pi Д В$	C	25806,6

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_e = 19$ прим.):

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

7.7 Визначення експлуатаційних витрат

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

Таблиця 7.11 – Розрахунок експлуатаційних витрат у споживача програмної продукції

Найменування статей витрат	Позначення	Сума витрат за варіантами, грн.	
		Базовий	Новий
1. Витрати на технічне обслуговування	Z_p	32208	13420
2. Витрати на електроенергію	$Z_{ел}$	5320	4560
3. Витрати на амортизацію	$Z_{ам}$	0	6352
Всього витрат за рік	I	37528	24332

Витрати на профілактичні роботи:

$$Z_p = T_p \cdot Z_2 \cdot (1 + 0,01 \cdot H_q) \cdot (1 + 0,01 \cdot H_c), \quad (7.23)$$

де: T_p – кількість годин обслуговування кожного комп'ютера за рік, год.;

Z_2 – заробітна плата обслуговуючого персоналу, грн/год.

Після купівлі нового програмного забезпечення кількість профілактичних годин робіт зменшилася з 240 годин на рік до 100 годин на рік, тому витрати на технічне обслуговування зменшилися з:

$$Z_{p \text{ баз}} = 240 \cdot 100 \cdot 1,1 \cdot 1,22 = 32208 \text{ грн},$$

до:

$$Z_{p \text{ нов}} = 100 \cdot 100 \cdot 1,1 \cdot 1,22 = 13420 \text{ грн}.$$

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 ($C_{ел}$):

$$Z_{ел} = П_{ел} \cdot T_p \cdot Ц_{ел} \quad (7.24)$$

$$Z_{ел баз} = 0,35 \cdot 4000 \cdot 3,8 = 5320 \text{ грн.}$$

$$Z_{ел нов} = 0,3 \cdot 4000 \cdot 3,8 = 4560 \text{ грн.}$$

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

Таблиця 7.12 – Розрахунок амортизаційних відрахувань

Групи основних фондів	Норма амортизації %	Балансова вартість, грн., за варіантами		Сума відрахувань, грн за варіантами	
		Базовий	Новий	Базовий	Новий
Програмна продукція	25	–	25807	–	6451,75
Всього відрахувань	-	–	25807	–	6451,75

7.8 Визначення економічної ефективності програмної продукції

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

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

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

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

$$E_e = (21505,5 - 14337) \cdot 19 - (0,05 \cdot 1408000 + 0,5 \cdot 114885 + 0,25 \cdot 105190 + 0,2 \cdot 97500 + 0,1 \cdot 19000) \cdot 3/12 = 92317 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у виробника програмної продукції:

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

де: K_p^* – балансова вартість основних фондів розробника.

$$T_e = \frac{1744575}{(21505,5 - 14337) \cdot 19 \cdot 12 / 3} = 3,2 \text{ років.}$$

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.	19
2. Повна собівартість розробленої програми	Грн.	14337
3. Ціна розробленої програми	Грн.	21505,5
4. Плановий прибуток від реалізації розробленої програми	Грн.	7168,5
5. Рентабельність програмної продукції	%	50
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	1744575
7. Загальний прибуток від реалізації програмної продукції	Грн.	136201,5
8. Величина економічного ефекту при виготовлені програмної продукції	Грн.	92317
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років.	3,2
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	25807
11. Величина економічного ефекту у користувача програмної продукції	Грн.	6744
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	1,95

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

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

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

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

$$E_{cn} = (37528 - 24332) - 0,25 \cdot 25807 = 6744 \text{ грн.}$$

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

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

$$T_{cn} = \frac{25807}{37528 - 24332} = 1,95 \text{ року.}$$

7.9 Висновки

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

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		89

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Законом України “Про охорону праці” [1] регламентуються загальні положення державної політики в галузі охорони праці, а конкретизуються ці положення нормативно-правовими актами про охорону праці, зокрема Наказом Міністерства соціальної політики України 14.02.2018 № 207 [5], який зареєстровано в Міністерстві юстиції України 25 квітня 2018 р. за №508/31960 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями», НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин», та ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» [2].

Програмісти у процесі роботи мають негативний вплив на органи зору, а також мають значну розумову напругою і нервово-емоційне навантаження. Руки (суглоби пальців та м'язи рук) при роботі з клавіатурою мають теж істотне навантаженням [2]. До шкідливих факторів, які впливають на робітників галузі інформаційних технологій (ІТ) спеціалісти відносять високочастотні електромагнітні коливання (випромінювання) роботи апаратної частини ЕОМ та виділення шкідливих газів.

Ці шкідливі фактори можуть привести до професійних захворювань.

При розгляді шкідливих чинників роботи програмістів та інших спеціалістів ІТ будемо керуватись наступними нормативно-правовими актами: «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98 [2], та «Правила охорони праці під час експлуатації електронно-обчислювальних машин» НПАОП 0.00-1.28-10,

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

Умови праці програміста вулячають наступні фактори:

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

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

8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

До фізичних факторів при роботі на комп'ютері можемо віднести:

- підвищену й знижену температуру повітря;
- підвищену й знижену вологість повітря;
- недостатню освітленість робочого місця;
- перевищуючі припустимі норми шуму;
- підвищений рівень іонізуючого випромінювання;
- підвищений рівень електромагнітних полів;
- підвищений рівень статичної електрики;
- небезпеку враження електричним струмом;
- бляклість екрана дисплея;

до хімічних можемо віднести:

– виникнення, у результаті іонізації повітря при роботі комп'ютера, активних часток.

До психофізіологічні можемо віднести:

- розумова напруга;
- втрата реальності;
- виникнення залежності;
- нервово-емоційні перевантаження;
- перенапруга зорового аналізатора.

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

До біологічних можемо віднести:

– бактеріологічну небезпеку пов'язану з наявністю місць з сприятливим середовищем їх розмноженням (наприклад клавіатура).

8.3 Пропозиції щодо підвищення працездатності ІТ-фахівця

Практичне значення заходів щодо підвищення працездатності впливає із закономірностей її динаміки і зводиться ось до чого:

- збільшення фази стійкого стану у фонді робочого часу;
- прискорення процесу впрацювання;
- віддалення фази розвитку втоми;
- забезпечення високої продуктивності праці за нормальних фізіологічних затрат.

Комплекс заходів щодо підвищення і збереження працездатності працівників на оптимальному рівні реалізується на техніко-організаційному, соціально-економічному, санітарно-гігієнічному, медико-біологічному, психологічному напрямках.

Вагомим фактором високої працездатності і продуктивності праці є оптимізація трудових навантажень на основі механізації і автоматизації виробничих процесів, удосконалення технології, скорочення і ліквідації важкої ручної праці. Доведено, що при правильній організації праці на легких роботах спостерігається найбільша тривалість фази стійкого стану, а на важких роботах вона нетривала.

Високий рівень працездатності безпосередньо залежить від умов праці, оскільки поліпшення їх супроводжується зменшенням енергетичних затрат організму на подолання несприятливого впливу факторів виробничого середовища.

Важливим напрямком підвищення працездатності працюючих є ритмізація трудових процесів, оптимізація темпу роботи, а також раціоналізація трудових рухів на фізіологічній основі, що сприяє формуванню і закріпленню робочих

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

динамічних стереотипів, а отже зменшенню м'язових і вольових зусиль. Ритмічна робота підвищує функціональні можливості організму, сприяє його тренуваності і забезпечує економізацію енергетичних затрат. [1]

Багатьом програмістам постійно доводиться працювати з великою кількістю програм одночасно. Часте перемикання туди-сюди між IDE та довідкою суттєво зменшує продуктивність фахівця. Однак вирішення цієї проблеми досить просте та очевидне: встановлення більшої кількості моніторів.

Оптимальним варіантом є два монітори. Все ж таки це найпростіший з апаратної точки зору варіант. Крім того, якби їх було більше, то ними було б важче керувати, та й столі просто не вистачить місця на ще один монітор. Але тут ще залежить розміру моніторів. Є системи із 4 або 6 відносно невеликими екранами, які кріпляться на кронштейні. Але оптимальним є два 27-дюймові монітори, на яких все добре видно, особливо коли працювати доводиться в основному з текстом [3].

8.4 Розрахункова частина

Завдання: розрахувати *штучне освітлення робочого приміщення*.

Початкові дані: ширина робочого приміщення: 2 м.; довжина – 3,5 м.; висота – 3 м.

Розрахунок штучного освітлення проведемо за методом коефіцієнта використання світлового потоку.

Для того, щоб визначити потрібну кількість світильників, які повинні забезпечити нормований рівень освітленості, визначимо світловий потік, що падає на робочу поверхню за формулою:

$$F=ESKZ/n,$$

де: F – світловий потік, що розраховується, Лм;

E – нормована мінімальна освітленість, Лк; $E = 300$ Лк;

S – площа освітлюваного приміщення (у нашому випадку $S=2 \times 3,5 = 7 \text{ м}^2$);

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

Z – відношення середньої освітленості до мінімальної (зазвичай приймається рівним 1.1... 1.2, в нашому випадку $Z = 1,1$);

K – коефіцієнт запасу, що враховує зменшення світлового потоку лампи в результаті забруднення світильників в процесі експлуатації (його значення залежить від типу приміщення і характеру робіт, що проводяться в ньому, в нашому випадку $K = 1,5$);

n – коефіцієнт використання світлового потоку, (відношення світлового потоку, що падає на розрахункову поверхню, до сумарного потоку всіх ламп і обчислюється в долях одиниці; залежить від характеристик світильника, розмірів приміщення, забарвлення стін і стелі, що характеризуються коефіцієнтами відбиття від стін ($\rho_{стін}$) і стелі ($\rho_{стелі}$), значення коефіцієнтів дорівнюють $\rho_{стін} = 50\%$ і $\rho_{стелі} = 50\%$ [6].

Обчислимо індекс приміщення за формулою:

$$i = S / (h(A+B)),$$

де: S – площа приміщення, $S = 7 \text{ м}^2$;

h – розрахункова висота підвісу, $h = 3 \text{ м}$;

A – ширина приміщення, $A = 2 \text{ м}$;

B – довжина приміщення, $B = 3,5 \text{ м}$.

Підставимо всі значення у формулу та визначимо індекса приміщення:
 $i = 0,57$.

Знаючи індекс приміщення, за знаходимо $n = 0,29$ (з табличних даних коефіцієнтів використання світлового потоку (n) світильників відповідного типу). Підставимо всі значення у формулу, визначимо світловий потік: $F = 7533 \text{ Лм}$.

Для штучного освітлення приміщення використовуються *LED панель MAXUS ASSISTANCE PRO 80W 5000K WHITE (M1052480531)*, світловий потік яких $F_n = 8000 \text{ Лм}$.

Число світильників визначається по формулі:

$$N = F / F_n$$

де: F – світловий потік,

F_n – світловий потік одного світильника.

$$N = 7533 / 8000 = 0,94 \text{ шт.}$$

Приймаємо необхідну кількість світильників 1 шт.

8.5 Висновки до розділу

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

З цих міркувань було здійснено аналіз приміщення, призначеного для праці програмістів, проведено розгляд небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи. Виконано розрахунок захисного штучного освітлення. Розроблено заходи з охорони праці.

Список використаних джерел інформації

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

2. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин: ДСанПІН 3.3.2.007-98. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/v0007282-98>

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

4. Наказ Міністерства соціальної політики України 14.02.2018 № 207 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/z0508>

5. Постанова № 42 від 01.12.1999 Головного державного санітарного лікаря України «Санітарні норми мікроклімату виробничих приміщень ДСН 3.3.6.042-99. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/va042282-99>

6. Оришака, О. В. Основи охорони праці: навч. посіб. / О. В. Оришака, Г. П. Горбачова, К. М. Марченко; М-во освіти і науки України, Центральноукраїн. нац. техн. ун-т. – Кропивницький: ЦНТУ, 2022. – 175 с. – Режим доступу до ресурсу: <http://dspace.kntu.kr.ua/jspui/handle/123456789/12161> (дата звернення: 16.06.2023).

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		95

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи надання доступу до мережі Internet сервіс провайдера.

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 надання доступу до мережі Internet сервіс провайдера.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем надання доступу до мережі Internet сервіс провайдера.

– Досліджена система надання доступу до мережі Internet сервіс провайдера.

– На основі отриманих результатів досліджень створена програмна реалізація системи надання доступу до мережі Internet сервіс провайдера.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання надання доступу до мережі Internet сервіс провайдера.

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-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		96

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 РНР, Perl. 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 CAST-128.

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 6744 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 1,95 роки.

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Штогрін Б.В. Дослідження та програмна реалізація системи надання доступу до мережі Internet сервіс провайдера // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2023.
2. Оліфер В.Г. Комп'ютерні мережі. Принципи, технології, протоколи. Підручник / В.Г. Оліфер, Н.А.Оліфер. – [5-е вид.]. – 2016. – 944 с.
3. Е. Таненбаум, Д. Уезеролл «Комп'ютерні мережі». – [5-е вид.]. – 2016. – 960 с.
4. Wendell Odom. «CCNA 200-301 Official Cert Guide, Volume 1». Cisco Press. 2020. – 848 p.
5. Wendell Odom. «CCNA 200-301 Official Cert Guide, Volume 2 Premium Edition eBook and Practice Test». Cisco Press. 2020. – 624 p.
6. Scott Jernigan «CompTIA Network+ Certification All-in-One Exam Guide, Eighth Edition». 2022. – 976 p.
7. Doug Lowe «Networking For Dummies 12th Edition». 2020. – 480 p.
8. Ramon Nastase «Computer Networking: The Beginner's guide for Mastering Computer Networking, the Internet and the OSI Model». 2018. – 186 p.
9. Russ White & Ethan Banks «Computer Networking Problems and Solutions: An Innovative Approach to Building Resilient, Modern Networks». 2017. – 832 p.
10. 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.
11. Smirnova, T., Gnatyuk, S., Yudin, O., Sydorenko, V., Polozhentsev, A., «The Model for Calculating the Quantitative Criteria for Assessing the Security Level

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		98

of Information and Telecommunication Systems». *CEUR Workshop Proceedings* Volume 3156, 2022, Pages 390-399.

12. Smirnova T., Gnatyuk S., Berdibayev R., Avkurova Zh., Iavich M. «Cloud-Based Cyber Incidents Response System and Software Tools». *Communications in Computer and Information Science*, 2021, vol 1486. Springer, Cham. pp 169-184.

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

14. 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.

15. 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.

16. 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.

17. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». *CEUR Workshop Proceedings* Volume 2616, 2020, Pages 125-136.

18. 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.

19. 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.

20. 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.

21. 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.

22. 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.

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

24. 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.

25. 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.

26. Smirnov, O., Kuznetsov, A., Kiian, A., Zamula, A., Rudenko, S., Hryhorenko, V., «Variance Analysis of Networks Traffic for Intrusion Detection in

Smart Grids», 2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS), Kyiv, Ukraine April 17-19, 2019 P. 353-358.

27. 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.

28. 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.

29. 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.

30. Аль-Мудхафар Акіл Абдулхуссейн М., Смірнова Т.В., Буравченко К.О., Смірнов О.А. «Метод оцінки та підвищення користувальницького досвіду абонентів в програмно-конфігурованих мережах на основі використання машинного навчання». *Сучасні інформаційні системи*, 2023, том 7, № 2, С. 49-56.

31. Смірнова Т.В., Гнатюк С.О., Сидоренко В.М., Юдін О.Ю., Сидоренко С.Ю., «Модель визначення критичності галузевих інформаційно-телекомунікаційних систем». *Проблеми інформатизації та управління*, № 2(70). 2022. С. 28-37.

32. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Смірнов С.А., Поліщук Л.І., «Дослідження стійкості до диференціального криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 3(69). С. 93-98.

33. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Поліщук Л.І., Смірнов С.А. «Дослідження статистичної стійкості та швидкісних характеристик

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		101

запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Вісник Хмельницького національного університету. Серія: «Технічні науки», № 2 (307). С. 46-52. 2022.*

34. Смірнов О.А., Смірнова Т.В., Константинова Л.В., Смірнов С.А., Якименко Н.М., «Дослідження стійкості до лінійного криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку, 2022, № 1(67). С. 84-89.*

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

36. Смірнов О.А., Усік П.С., Миронець І.В., Буравченко К.О., Якименко Н.М. «Метод підвищення ефективності розподіленої обробки даних у комп'ютерних системах операторів стільникового зв'язку» *Вісник Черкаського державного технологічного університету. Технічні науки. №4. С. 103-110. 2020.*

37. 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.*

38. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В., Поліщук Л.І. Інформаційна безпека в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2020. – 294 с.

39. 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.*

40. Смірнов О.А., Дреєва Г.М., Дреєв О.М., Смірнова Т.В. «Фрактальний аналіз генератора самоподібного трафіку на основі ланцюга Маркова».

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		102

Центральноукраїнський науковий вісник. Технічні науки. № 2(33). с. 161-172, 2019.

41. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В. Поліщук Л.І. Проектування комп'ютерних систем та мереж. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2019. – 264 с.

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

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

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

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

46. Смірнов О.А., Смірнов С.А., Поліщук Л.І., Смірнова Т.В., Коноплицька-Слободенюк О.К. Метод формування антивірусного захисту даних з використанням безпечної маршрутизації метаданих. Кібербезпека: освіта, наука, техніка. – Том 3 № 3. – Київ: КУ ім. Бориса Грінченка. – 2019. – С. 63-87.

47. Смірнов О.А., Гнатюк С.О., Кавун С.В., Терейковський І.А., Жмурко Т.О., Смірнов С.А., Коваленко А.С. Основи безпеки в комп'ютерних

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		103

мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2018. – 177 с.

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

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

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

51. Смірнов О.А., Смірнов С.А., Дідик А.К., Дреєв А.М. Спосіб контролю ліній зв'язку телекомунікаційної системи антивірусу. Спосіб контролю ліній зв'язку телекомунікаційної системи антивірусу. Збірник наукових праць Харківського університету Повітряних Сил. Випуск 2 (47). – Харків: ХУПС. - 2016. - С. 121-127.

					ВКРМ-123.23.0024.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		104

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-123.23.0024.00.00.ТЗ		
Вим.	Арк.	№ документа	Підпис	Дата			
Розробив	Штогрін Б.В.				Літ.	Аркуш	Аркушів
Перевірів	Кислун О.А.						
Н. Контр.	Коваленко А.С.				ЦНТУ КІ-22М-1		
Затв.	Смірнов О.А.						

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи надання доступу до мережі Internet сервіс провайдера.

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. № 34-13 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 надання доступу до мережі Internet сервіс провайдера.

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-123.23.0024.00.00.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 надання доступу до мережі Internet сервіс провайдера;
- 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-123.23.0024.00.00.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 PHP, Perl.

					ВКРМ-123.23.0024.00.00.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 пропозиції щодо підвищення працездатності ІТ-фахівця.

					ВКРМ-123.23.0024.00.00.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 – 104 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 21.12.2023 р.

					ВКРМ-123.23.0024.00.00.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

_____ Кислун О.А.

***Дослідження та програмна реалізація
системи надання доступу до мережі Internet сервіс провайдера***

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 33

Літера: РП

Кропивницький – 2023 року

ВИХІДНИЙ КОД МОДУЛЯ АДМІНІСТРУВАННЯ

```

<?php
//Штогрін Богдан Васильович
// ЦНТУ, кафедра КБПЗ, КІ-22М-1, 2023

    $path=getenv("DOCUMENT_ROOT");
    $path.="/MONITORING /"; // шлях до папки зі скриптами
    include( "$path"."common.php" ); // підключення модулю common.php
    include( "$path"."global.php" ); // підключення модулю global.php
    Top("./", "сторінка адміністратора");
    ConnectToBase($conf{'host'}, $conf{'username'}, $conf{'password'},
    $conf{'database'});
    $CurrentDateDB = GetCurrentDateDB();
    if ($action=="search") // виклик пошуку?
    {
    search();
    }
    elseif ($action=="show") // виклик відображення інтерфейсу
    {
    ShowUserAdmin($ip,$month);
    }
    elseif ($action=="incbalance") // виклик відображення балансу
    {
    IncBalance ();
    ShowUserAdmin($ip,$month);
    }
    elseif ($action=="update") // виклик оновлення
    {
    UpdateUser ();
    ShowUserAdmin($ip,$month);
    }
    elseif ($action=="delete") // виклик видалення
    {
    if ($is_js_confirmed)
    {
    DeleteUser ();
    }
    ShowSearchForm(GetMonths());
    }
    elseif ($action=="adduserform")
    {
    AddUserForm ( );
    }
    elseif ($action=="adduser") // виклик додавання користувача
    {
    AddUser( );
    ShowSearchForm(GetMonths());
    }
    elseif ($action=="addgroup")
    {
    AddGroup ();
    EditGroup ();
    }elseif ($action=="editgroup")
    {
    EditGroup ();
    }
    elseif ($action=="delgroup")
    {
    DelGroup ();
    EditGroup ();
    }
    elseif ($action=="updategroup")
    {
    UpdateGroup ();
    EditGroup ();
    }
    elseif ($action=="stat")

```

```
{
  ShowStat ();
}
elseif ($action=="deleteold")
{
  DeleteOld ();
} else {
ShowSearchForm(GetMonths ());
}
Footer();
?>
```

К6П3 - 2023

ВИХІДНИЙ КОД WWW-ІНТЕРФЕЙСУ

```
<?php
//Штогрін Богдан Васильович
// ЦНТУ, кафедра КБПЗ, КІ-22М-1, 2023
include( "../common.php" ); // підключення модулю common.php
include( "../global.php" ); // підключення модулю global.php
// початок формування HTML коду
Тор("../", "сторінка користувача.");
// підключення та робота з БД, формування HTML коду
ConnectToBase($conf{'host'}, $conf{'username'}, $conf{'password'},
              $conf{'database'});
// Дані користувачів, формування HTML коду
ShowUser(ResolveUser());
// кінець формування HTML коду
Footer();
?>
```

КБПЗ - 2023

ФАЙЛ MONITORING.PL - ВИХІДНИЙ КОД МОДУЛЯ ЗЧИТУВАННЯ ТРАФІКА

```
#!/usr/bin/perl
# Diplom
# Штогрін Богдан Васильович
# ЦНТУ, кафедра КБПЗ, КІ-22М-1, 2023
use DBI;
use Common;
LoadConfig();
GetDate();
ConnectToDB();
TestNextMonth();
PutTrafToDBOutput();
PutTrafToDBForward();
SubtractAbon();
TurnOffAccess();
TurnOnAccess();
KillUserOutput();
KillUserForward();
Disconnect();
```

КБПЗ - 2023

ФАЙЛ COMMON.PM - ПАКЕТНИЙ ФАЙЛ ПРОЕКТУ

```

package Common;
BEGIN {
//Штогрін Богдан Васильович
// ЦНТУ, кафедра КБПЗ, КІ-22М-1, 2023
    use Exporter ();
    @ISA = 'Exporter';
    @EXPORT = qw(&SubtractAbon &ConvertMoneyToDB &TurnOnAccess &TurnOffAccess
&SubtractMoney &GetDate &PutTrafToDBOutput &PutTrafToDBForward &LoadConfig
&ConnectToDB &TestNextMonth &KillUserOutput &KillUserForward &Disconnect);
}
// завантаження конфігурації
sub LoadConfig
{
for $file("./Monitoring.conf")
    {
        // повернення коду читання файлу
        unless ($return = do $file) {
            warn "couldn't parse $file: $@" if $@;
            warn "couldn't do $file: $!" unless defined $return;
            warn "couldn't run $file" unless $return;
        }
    }
}

// читання таблиць БД
sub TurnOffAccess
{
    my $query_counter = ("SELECT `ip` FROM `counter` WHERE
        `moneymonth`='n' AND `yearmonth`='$date'");
    my $sth_counter = $dbh -> prepare($query_counter);
    $sth_counter -> execute;
    $sth_counter -> bind_columns(undef, \($ip));
    while ($sth_counter -> fetch)
    {
        my $query = ("UPDATE `user` SET access = 'n' WHERE `ip` = '$ip' AND
            `access`='y'");
        my $sth = $dbh -> prepare($query);
        $sth -> execute;
        $sth -> finish;
    }
    $sth_counter -> finish;
my $query = ("UPDATE `user` SET access = 'n' WHERE `balance`<=0 AND
    `access`='y'");
    my $sth = $dbh -> prepare($query);
    $sth -> execute;
    $sth -> finish;
}
sub TurnOnAccess
{
    my $query_counter = ("SELECT `ip` FROM counter WHERE
        `moneymonth`='y' AND `yearmonth`='$date'");
    my $sth_counter = $dbh -> prepare($query_counter);
    $sth_counter -> execute;
    my ($ip);
    $sth_counter -> bind_columns(undef, \($ip));
    while ($sth_counter -> fetch)
    {
        my $query = ("UPDATE `user` SET access = 'y' WHERE `balance` > 0 AND
`ip` = '$ip' AND `access`='n'");
        my $sth = $dbh -> prepare($query);
        $sth -> execute;
        $sth -> finish;
    }
    $sth_counter -> finish;
}

// Запис до таблиць БД

```

```

sub ConvertMoneyToDB
{
    $money = $_[0];
    $newmoney = $money*1000000000;
return $newmoney; // повернення коду затрачених коштів
}
sub SubtractAbon
{
    $ip = $_[0];
    # $balancelimit = ConvertMoneyToDB(100);
    my $query_user = ("SELECT `ip`,`group`,`balance` FROM user");
    #print "$query_user\n";
    my $sth_user = $dbh -> prepare($query_user);
    $sth_user -> execute;
    $sth_user -> bind_columns(undef,\($ip,$group,$balance));
    while( $sth_user -> fetch )
    {
        my $query_counter = ("SELECT `moneymonth`,`counter` FROM `counter`
                                WHERE `yearmonth`='$date' AND `ip`='$ip'");
        my $sth_counter = $dbh -> prepare($query_counter);
        $sth_counter -> execute;
        my ($moneymonth,$counter);
        $sth_counter -> bind_columns(undef,\($moneymonth,$counter));
        $sth_counter -> fetch;
        $sth_counter -> finish;
        if ($moneymonth eq 'n')
        {
            my $query_group = ("SELECT `trafcost`,`abon`,`abontraf` FROM
                                `group` WHERE `group`='$group'");
            my $sth_group = $dbh -> prepare($query_group);
            $sth_group -> execute;
            my ($trafcost,$abon,$abontraf);
            $sth_group -> bind_columns(undef,\($trafcost,$abon,$abontraf));
            $sth_group -> fetch;
            $sth_group -> finish;
            if ($balance >= ConvertMoneyToDB($abon))
            {
                my $MoneyForAbon = ConvertMoneyToDB($abon)-1;
                my $query_abon = ("UPDATE `user` SET balance = balance -
                                    $MoneyForAbon WHERE `ip`='$ip'");
                my $sth_abon = $dbh -> prepare($query_abon);
                $sth_abon -> execute;
                $sth_abon -> finish;
            }
        }
    }
    $sth_user -> finish;
}
sub SubtractMoney
{
    $ip = $_[0];
    $traffic = $_[1];
    my $query_user = ("SELECT `group` FROM user WHERE `ip`='$ip'");
    #print "$query_user\n";
    my $sth_user = $dbh -> prepare($query_user);
    $sth_user -> execute;
    my ($group);
    $sth_user -> bind_columns(undef,\($group));
    $sth_user -> fetch;
    $sth_user -> finish;
    my $query_group = ("SELECT `trafcost`,`abontraf` FROM `group`
                                WHERE `group`='$group'");
    my $sth_group = $dbh -> prepare($query_group);
    $sth_group -> execute;
}

```

```

my ($trafcost, $abontraf);
$sth_group -> bind_columns(undef, \($trafcost, $abontraf));
$sth_group -> fetch;
$sth_group -> finish;
    my $query_counter = ("SELECT `moneymonth`,`counter` FROM `counter`
                          WHERE `yearmonth`='$date' AND `ip`='$ip'");
    my $sth_counter = $dbh -> prepare($query_counter);
$sth_counter -> execute;
my ($moneymonth, $counter);
$sth_counter -> bind_columns(undef, \($moneymonth, $counter));
$sth_counter -> fetch;
$sth_counter -> finish;
if ($counter > $abontraf*1048576)
{
    my $ByteCost = $trafcost/(100*1024*1024);
    my $MoneyToSubtract = ConvertMoneyToDB($ByteCost*$traffic);
my $query_balance = ("UPDATE `user` SET balance = balance - $MoneyToSubtract
                     WHERE `ip`='$ip'");
    my $sth_balance = $dbh -> prepare($query_balance);
    $sth_balance -> execute;
    $sth_balance -> finish;
}
}
sub KillUserForward
{
    my $query = ("SELECT ip,access FROM user");
    my $sth = $dbh -> prepare($query);
    $sth -> execute;
    my ($ip, $access);
    $sth -> bind_columns(undef, \($ip, $access));
    while($sth -> fetch)
    {
        #print "$ip $access\n";
        if ($access eq "n")
        {
            my $AllreadyOff='n';
            open (KILLHFORWARD, "iptables -L FORWARD -n |");
            while (my $s=<KILLHFORWARD>)
            {
                if (($.>2))
                {
                    @ar = split /\s+/, $s;
                    #print "$ar[0] $ar[3]\n";
                    if (($ar[0] eq "REJECT") && ($ar[3] eq "$conf{'network'}"."$ip"))
                    {
                        $AllreadyOff = 'y';
                    }
                }
            }
            close(KILLHFORWARD);
            if ($AllreadyOff eq 'n')
            {
                $string = "iptables -I FORWARD -s ".$conf{'network'}.$ip." -j REJECT";
                system($string);
            }
            else {
                my $AllreadyOff='n';
                open (KILLHFORWARD, "iptables -L FORWARD -n |");
                while (my $s=<KILLHFORWARD>)
                {
                    if (($.>2))
                    {
                        @ar = split /\s+/, $s;
                        #print "$ar[0] $ar[3]\n";
                        if (($ar[0] eq "REJECT") && ($ar[3] eq
"$conf{'network'}"."$ip"))
                        {
                            $AllreadyOff = 'y';
                        }
                    }
                }
            }
        }
    }
}

```

```

        }
    }
    close(KILLHFORWARD);
    if ($AllreadyOff eq 'y')
    {
$string = "iptables -D FORWARD -s ".$conf{'network'}. $ip." -j REJECT";
        system($string);
    }
}
}
$sth -> finish;
}
sub KillUserOutput
{
    my $query = ("SELECT ip,access FROM user");
    my $sth = $dbh -> prepare($query);
    $sth -> execute;
    my ($ip,$access);
    $sth -> bind_columns(undef,\($ip,$access));
    while($sth -> fetch)
    {
        print "$ip $access\n";
        if ($access eq "n")
        {
            my $AllreadyOff='n';
            open (KILLHOUTPUT, "iptables -L OUTPUT -n |");
            while (my $s=<KILLHOUTPUT>)
            {
                if (($.>2))
                {
                    {
                        @ar = split /\s+/, $s;
                        #print "$ar[0] $ar[3]\n";
                    if (($ar[0] eq "REJECT") && ($ar[4] eq "$conf{'network'}. "$ip"))
                    {
                        $AllreadyOff = 'y';
                    }
                }
            }
            close(KILLHOUTPUT);
            if ($AllreadyOff eq 'n')
            {
my $string = "iptables -I OUTPUT -d ".$conf{'network'}. $ip." -p tcp --sport 3128
                -j REJECT";
                system($string);
            }
            } else {
                my $AllreadyOff='n';
                open (KILLHOUTPUT, "iptables -L OUTPUT -n |");
                while (my $s=<KILLHOUTPUT>)
                {
                    if (($.>2))
                    {
                        {
                            @ar = split /\s+/, $s;
                            #print "$ar[0] $ar[3]\n";
                        if (($ar[0] eq "REJECT") && ($ar[4] eq "$conf{'network'}. "$ip"))
                        {
                            $AllreadyOff = 'y';
                        }
                    }
                }
            }
            close(KILLHOUTPUT);
            if ($AllreadyOff eq 'y')
            {
                my $string = "iptables -D OUTPUT -d ".$conf{'network'}. $ip." -
                p tcp --sport 3128 -j REJECT";
                system($string);
            }
        }
    }
}
}

```

```

        $sth -> finish;
    }

    sub ConnectToDB
    {
        $datasource = "DBI:mysql:". $conf{ 'DBName' }. ":". $conf{ 'host' };
        $dbh = DBI -> connect($datasource, $conf{ 'username' }, $conf{ 'password' });
    }

    sub GetDate
    {
        open (DATE_H, "date +%y%m |");
        $date = <DATE_H>;
    }

    sub TestNextMonth
    {
        my $query = ("SELECT ip FROM counter WHERE yearmonth='$date'");
        my $sth = $dbh -> prepare($query);
        $sth -> execute;
        my $numrows = $sth -> rows;
        $sth -> finish;
        if ($numrows == 0)
        {
            for (my =1;$i<=254;$i++)
            {
                my $query = ("INSERT INTO counter (ip,yearmonth) VALUES
($i,'$date')");
                my $sth = $dbh -> prepare($query);
                $sth -> execute;
                $sth -> finish;
            }
        }
    }

    sub PutTrafToDBOutput
    {
        $count=0;
        open (FH, "iptables -L -Z OUTPUT -nvx |");
        print "$date\n";
        while(my $s=<FH>)
        {
            if (($.>2))
            {
                # $s =~ s/\s+/_/g;
                @ar = split /\s+/, $s;
                if (($ar[9] =~ /$conf{ 'network' }\d+/) && ($ar[3] ne "REJECT"))
                {
                    @ip = split /\./, $ar[9];
                    if ($ar[2] ne "0")
                    {
                        if (($ip[3] >= 2) && ($ip[3] <= 254))
                        {
                            my $query = ("UPDATE counter SET counter = counter + $ar[2] WHERE
ip='$ip[3]' AND yearmonth='$date'");
                            my $sth = $dbh -> prepare($query);
                            $sth -> execute;
                            $sth -> finish;
                            SubtractMoney("$ip[3]","$ar[2]");
                        }
                        $count+=$ar[2];
                    }
                }
            }
        }
        print "\nOverall OUTPUT: $count\n";
        close(FH);
    }

    sub PutTrafToDBForward
    {
        $count=0;
        open (FH, "iptables -L -Z FORWARD -nvx |");
    }

```

```

print "$date\n";
while(my $s=<FH>)
{
    if (($.>2))
    {
        # $s =~ s/\s+/_/g;
        @ar = split /\s+/, $s;
        if (($ar[9] =~ /\$conf{'network'}\d+/)&&($ar[3] ne "REJECT"))
        {
            @ip = split /\./, $ar[9];
            if ($ar[2] ne "0")
            {
                if (($ip[3] >= 2) && ($ip[3] <= 254))
                {
                    my $query = ("UPDATE counter SET counter = counter + $ar[2] WHERE
ip='$ip[3]' AND yearmonth='$date'");
                    my $sth = $dbh -> prepare($query);
                    $sth -> execute;
                    $sth -> finish;
                    SubtractMoney("$ip[3]","$ar[2]");
                }
                $count+=$ar[2];
            }
        }
    }
}
print "\nOverall FORWARD: $count\n";
close(FH);
}
sub Disconnect
{
    $dbh -> disconnect;
}
return 1; // повернення коду
END { }

```

ВИХІДНИЙ КОД ФАЙЛА COMMON.PHP

```

<?php
//Шторпін Богдан Васильович
// ЦНТУ, кафедра КБПЗ, КІ-22М-1, 2023
// функція підключення та роботи з БД
function ConnectToBase($host,$username,$password,$database)
{
    global $link;
    if (!($link = mysql_pconnect( $host, $username, $password)))
        {
            echo ("<br>Соединение с базой не установлено<br>");
            exit();
        }
// вибор БД
    if (!mysql_select_db($database, $link))
        {
            echo ( "Помилка при виклику БД. ".mysql_errno($link)."
                ".mysql_error($link) );
            exit();
        }
}

// Функція відображення форми пошуку користувачів
function ShowSearchForm($month)
{
    ?>
<table cellpadding="0" cellspacing="0" border="0">
<tr><td>
<table cellpadding="0" cellspacing="0" border="0">
<tr class="edge"><td>
    <table cellpadding="3" cellspacing="1" border="0">
    <form method="GET" action=?echo ("$_PHP_SELF");?>
    <tr class="even"><td height=30>
        Login:
    </td>
    <td bgcolor="#ffffff">
        <input type="text" name="login">
    </td>
    </tr>
    <tr class="even"><td height=30>
        IP:
    </td>
    <td bgcolor="#ffffff">
        <input type="text" name="ip">
    </td>
    </tr>
    <tr class="even"><td height=30>
        Имя:
    </td>
    <td bgcolor="#ffffff">
        <input type="text" name="name">
    </td>
    </tr>
    <tr class="even"><td height=30>
        Адреса:
    </td>
    <td bgcolor="#ffffff">
        <input type="text" name="address">
    </td>
    </tr>
    <tr class="even"><td height=30>
        Доступ:
    </td>
    <td bgcolor="#ffffff">
        <input type="radio" name="access" value="y">Есть&nbsp;
        <input type="radio" name="access" value="n">Нет&nbsp;
        <input type="radio" name="access" value="-1" checked>-
    </td>

```

```

</tr>
<tr class="even"><td height=30>
    Місяць:
</td>
<td bgcolor="#ffffff">
    <select name="month">
<?
        $i=1;
        while ($month[$i])
        {
            if (GetCurrentDateDB()==$month[$i])
            {
                echo "<OPTION VALUE=\"\$month[$i]\"
SELECTED>".ConvertDateDB($month[$i])."</option>";
            } else {
                echo "<OPTION
VALUE=\"\$month[$i]\">".ConvertDateDB($month[$i])."</option>";
            }
            $i++;
        }
?>
    </select>
</td>
</tr>
<tr class="odd"><td></td><td height=30 align="left">
    <input type="submit" value="Искать">
</td></tr>
    <input type="hidden" name="action" value="search" >
</form>
</table>
</td></tr>
</table>
</td>
<td valign="top">
<table cellpadding="0" cellspacing="0" border="0" width="300">
<tr class="odd"><td height=40 align="right">
<a href="<? echo ("\$PHP_SELF"."?action=adduserform");?>">Додати нового
користувача</a>
</td></tr>
<tr class="odd"><td height=40 align="right">
<a href="<? echo ("\$PHP_SELF"."?action=editgroup");?>">Редагування груп</a>
</td></tr>
<tr class="odd"><td height=40 align="right">
<a href="<? echo ("\$PHP_SELF"."?action=stat");?>">Статистика</a>
</td></tr>
<tr class="odd"><td height=40 align="right">
<a href="<? echo ("\$PHP_SELF"."?action=deleteold");?>">Видалення інформації</a>
</td></tr>
</table>
</td></tr>
</table>
<?
}

// Функція видалення
function DeleteOld()
{
    global $delete, $link, $month;
    $month_ar = GetMonths();
?>
<table cellpadding="0" cellspacing="2" border="0">
<tr class="odd"><td>
<a href="<? echo ("\$PHP_SELF"."?action=");?>">Повернутися</a>
</td></tr>
<tr class="odd"><td>
Видалити інформацію о трафіке та платежах за:
</td></tr>

```

```

<tr class="odd"><td>
<form method="POST" action="<? echo ("$_PHP_SELF");?>">
    <select name="month">
<?
        $i=1;
        while ($month_ar[$i])
        {
            if (GetCurrentDateDB()!=$month_ar[$i])
            {
                echo "<OPTION
VALUE=\"$_month_ar[$i]\">".ConvertDateDB($month_ar[$i])."</option>";
            }
            $i++;
        }
?>
    </select>
</td></tr>
<tr><td>
<br>
        <input type="hidden" name="action" value="deleteold">
        <input type="hidden" name="delete" value="1">
        <input type="submit" value="Удалить">
</form>
</td></tr>
</table>
<?
    if (($delete)&&($month))
    {
        $query1="DELETE FROM `counter` WHERE `yearmonth`='$_month.'";
        $query2="DELETE FROM `money` WHERE `date` LIKE '$_month.%'";
        echo $query1."<br>".$query2;
        if (!( $result1 = mysql_query($query1, $link ) ))
        {
            echo ( "Помилка при виконанні $query1
".mysql_errno($link)." ".mysql_error($link) );
            exit();
        }
        if (!( $result2 = mysql_query($query2, $link ) ))
        {
            echo ( "Помилка при виконанні $query2
".mysql_errno($link)." ".mysql_error($link) );
            exit();
        }
    }
}

// Функція пошуку
function Search()
{
    global $name, $login, $ip, $address, $access, $month, $link, $sortBy;
    if ($ip != "")
    {
        $query_user = "SELECT `ip`, `login`, `access`, `name`, `address`,
`balance`, `group` FROM user WHERE `ip`='$ip'";
    }
    elseif ($login != "")
    {
        $query_user = "SELECT `ip`, `login`, `access`, `name`, `address`,
`balance`, `group` FROM user WHERE `login` like '%$login%'";
    } else {
        $query_user = "SELECT `ip`, `login`, `access`, `name`, `address`,
`balance`, `group` FROM user WHERE 1=1";
        if ($name != "") {$query_user.=" AND `name` like '%$name%'"; }
        if ($address != "") {$query_user.=" AND `address` like
'%$address%'";}
        if ($access !="-1") {$query_user.=" AND `access` like
'$access'";}
    }
    if ($sortBy=="ip_asc")

```

```

{
$query_user.=" ORDER BY ip ASC";
} elseif ($sortby=="ip_desc")
{
$query_user.=" ORDER BY ip DESC";
} elseif ($sortby=="login_desc")
{
$query_user.=" ORDER BY login DESC";
} elseif ($sortby=="login_asc")
{
$query_user.=" ORDER BY login ASC";
} elseif ($sortby=="group_desc")
{
$query_user.=" ORDER BY `group` DESC";
} elseif ($sortby=="group_asc")
{
$query_user.=" ORDER BY `group` ASC";
} elseif ($sortby=="access_desc")
{
$query_user.=" ORDER BY `access` DESC";
} elseif ($sortby=="access_asc")
{
$query_user.=" ORDER BY `access` ASC";
} elseif ($sortby=="balance_desc")
{
$query_user.=" ORDER BY balance DESC";
} elseif ($sortby=="balance_asc")
{
$query_user.=" ORDER BY balance ASC";
} elseif ($sortby=="address_desc")
{
$query_user.=" ORDER BY address DESC";
} elseif ($sortby=="address_asc")
{
$query_user.=" ORDER BY address ASC";
} elseif ($sortby=="name_desc")
{
$query_user.=" ORDER BY name DESC";
} elseif ($sortby=="name_asc")
{
$query_user.=" ORDER BY name ASC";
} else {
$query_user.=" ORDER BY ip ASC";
}
if ($sortby == "traffic_asc")
{
$query_counter = "SELECT `ip`, `counter`, `moneymonth` FROM `counter`
WHERE `yearmonth`='".$month.'" AND `counter` > 0 ORDER BY 'counter' ASC";
if (!( $result_counter = mysql_query( $query_counter, $link ) ))
{
echo ( "Помилка при виконанні $query_counter ".mysql_errno($link)."
".mysql_error($link) );
exit();
}
} elseif ($sortby == "traffic_desc")
{
$query_counter = "SELECT `ip`, `counter`, `moneymonth` FROM `counter`
WHERE `yearmonth`='".$month.'" AND `counter` > 0 ORDER BY 'counter' DESC";
if (!( $result_counter = mysql_query( $query_counter, $link ) ))
{
echo ( "Помилка при виконанні $query_counter
".mysql_errno($link)." ".mysql_error($link) );
exit();
}
}
}
if (!( $result_user = mysql_query( $query_user, $link ) ))
{
echo ( "Помилка при виконанні $query_user
".mysql_errno($link)." ".mysql_error($link) );
}
}

```

```

        exit ();
    }
    $backst = getenv("QUERY_STRING");
    $row_user = mysql_fetch_object( $result_user );
    if ( mysql_num_rows($result_user)==1 )
    {
        ShowUserAdmin($row_user -> ip,$month);
    } elseif (mysql_num_rows($result_user) > 1)
    {
        ?>
        <table cellpadding="0" cellspacing="0" border="0">
        <tr class="odd"><td>
        <a href="<? echo ("$_PHP_SELF"."?action=");?>">Вернуться</a>
        </td></tr>
        <tr class="edge"><td>
        <table cellpadding="2" cellspacing="1" border="0">
        <tr class="top" align="center">
        <td><a href=<?echo
        ("$_PHP_SELF"."?action=search&name=$name&login=$login&ip=$ip&address=$address&acc
        ess=$access&month=$month&"); if ($sortby=="ip_asc") { echo "sortby=ip_desc"; }
        else { echo "sortby=ip_asc"; }?>>IP</a></td>
        <td><a href=<?echo
        ("$_PHP_SELF"."?action=search&name=$name&login=$login&ip=$ip&address=$address&acc
        ess=$access&month=$month&"); if ($sortby=="login_asc") { echo
        "sortby=login_desc"; } else { echo "sortby=login_asc"; }?>>Login</a></td>
        <td><a href=<?echo
        ("$_PHP_SELF"."?action=search&name=$name&login=$login&ip=$ip&address=$address&acc
        ess=$access&month=$month&"); if ($sortby=="name_asc") { echo "sortby=name_desc";
        } else { echo "sortby=name_asc"; }?>>Имя</a></td>
        <td><a href=<?echo
        ("$_PHP_SELF"."?action=search&name=$name&login=$login&ip=$ip&address=$address&acc
        ess=$access&month=$month&"); if ($sortby=="group_asc") { echo
        "sortby=group_desc"; } else { echo "sortby=group_asc"; }?>>Группа</a></td>
        <td><a href=<?echo
        ("$_PHP_SELF"."?action=search&name=$name&login=$login&ip=$ip&address=$address&acc
        ess=$access&month=$month&"); if ($sortby=="address_asc") { echo
        "sortby=address_desc"; } else { echo "sortby=address_asc"; }?>>Адрес</a></td>
        <td><a href=<?echo
        ("$_PHP_SELF"."?action=search&name=$name&login=$login&ip=$ip&address=$address&acc
        ess=$access&month=$month&"); if ($sortby=="access_desc") { echo
        "sortby=access_asc"; } else { echo "sortby=access_desc"; }?>>Доступ</a></td>
        <td><a href=<?echo
        ("$_PHP_SELF"."?action=search&name=$name&login=$login&ip=$ip&address=$address&acc
        ess=$access&month=$month&"); if ($sortby=="traffic_desc") { echo
        "sortby=traffic_asc"; } else { echo "sortby=traffic_desc"; }?>>Трафик</a></td>
        <td><a href=<?echo
        ("$_PHP_SELF"."?action=search&name=$name&login=$login&ip=$ip&address=$address&acc
        ess=$access&month=$month&"); if ($sortby=="balance_asc") { echo
        "sortby=balance_desc"; } else { echo "sortby=balance_asc";
        }?>>Баланс (грн.)</a></td>
        <td>user.balance</td>
        <td>Абонентка</td>
        <td></td></tr>
        <?
        if (($sortby=="traffic_asc") || ($sortby=="traffic_desc"))
        {
            $i=1;
            mysql_data_seek( $result_user, 0);
            while (( $row_counter = mysql_fetch_object( $result_counter ))
            {
                $query_user = "SELECT `ip`, `login`, `access`, `name`, `address`,
                `balance`, `group` FROM `user` WHERE `ip`='".$row_counter -> ip.'"";
            if (!( $result_user = mysql_query($query_user, $link ) ))
            {
                echo ( "Помилка при виконанні $query_user ".mysql_errno($link)."
                ".mysql_error($link) );
                exit ();
            }
            $row_user = mysql_fetch_object( $result_user );

```

```

        if ($i%2) {$class="odd";} else {$class="even";}
        $i++;
        printf("<tr class=\"\$class\"><td>%s</td><td><a
href=\"\$PHP_SELF?action=show&ip=%s&month=%s\">%s</a></td><td><a
href=\"\$PHP_SELF?action=show&ip=%s&month=%s\">%s</a></td><td
align=\"center\">%s</td><td>%s</td><td align=\"center\">%s</td><td
align=\"right\">%s</td><td align=\"right\">%s</td><td align=\"right\">%s</td><td
align=\"center\">%s</td><td><a href=\"\$PHP_SELF?action=delete&ip=%s\"
onclick=\"return confirmLink(this, 'видалити користувача
`s`s`)\">>Видалити</a></td></tr>\n",

$row_user -> ip, $row_user -> ip, $month, $row_user -> login, $row_user -> ip,
$month, $row_user -> name, $row_user -> group, $row_user -> address, $row_user -
> access, number_format($row_counter -> counter,0, chr(44), "."),
ConvertMoneyFromDB($row_user -> balance), $row_user -> balance, $row_counter ->
moneymonth, $row_user -> ip, $row_user -> login);
    }
} else {
    $i=1;
    mysql_data_seek( $result_user, 0);
    while (( $row_user = mysql_fetch_object( $result_user ))
        {
            $query_counter = "SELECT counter, moneymonth FROM
counter WHERE `ip`=\".$row_user -> ip.\" AND `yearmonth`=\".$month;
            if (!( $result_counter =
mysql_query($query_counter, $link ) )
                {
                    echo ( "Помилка при виконанні
$query_counter ".mysql_errno($link)." ".mysql_error($link) );
                    exit();
                }
            $row_counter = mysql_fetch_object(
$result_counter );
            if ($i%2) {$class="odd";} else {$class="even";}
            $i++;
            printf("<tr class=\"\$class\"><td>%s</td><td><a
href=\"\$PHP_SELF?action=show&ip=%s&month=%s\">%s</a></td><td><a
href=\"\$PHP_SELF?action=show&ip=%s&month=%s\">%s</a></td><td
align=\"center\">%s</td><td>%s</td><td align=\"center\">%s</td><td
align=\"right\">%s</td><td align=\"right\">%s</td><td align=\"right\">%s</td><td
align=\"center\">%s</td><td><a href=\"\$PHP_SELF?action=delete&ip=%s\"
onclick=\"return confirmLink(this, 'видалити користувача
`s`s`)\">>Видалити</a></td></tr>\n",
                $row_user -> ip, $row_user -> ip, $month, $row_user ->
login, $row_user -> ip, $month, $row_user -> name, $row_user -> group, $row_user
-> address, $row_user -> access, number_format($row_counter -> counter,0,
chr(44), "."), ConvertMoneyFromDB($row_user -> balance), $row_user -> balance,
$row_counter -> moneymonth, $row_user -> ip, $row_user -> login);
        }
}
?>
</table>
</td></tr>
</table>
<?
} else
{
?>
<table cellpadding="0" cellspacing="0" border="0">
<tr class="odd"><td>
<a href="<? echo ("\$PHP_SELF"."?action=");?>">Повернутися</a>
</td></tr>
<tr class="odd"><td>
Нема користувачів, відповідних пошукові
</td></tr>
</table>
<?>
mysql_free_result( $result_user );
    if ($result_counter)

```

```
{
    mysql_free_result( $result_counter );
}
echo $query_user;
echo "<br>".$query_counter;
}

// Функція конвертування коштів
function ConvertMoneyFromDB($money)
{
    $NewMoney = number_format($money/1000000000, 2, chr(44), ".");
    return $NewMoney; // повернення коду коштів
}

// Функція конвертування коштів
function ConvertMoneyToDB($money)
{
    $NewMoney = $money*1000000000;
    return $NewMoney; // повернення коду коштів
}

// Функція конвертування дати у БД
function ConvertDateDBMoney($dateDB)
{
    $date_ar = explode(" ", $dateDB);
    $time = $date_ar[1];
    $year = substr($date_ar[0], 0, 2);
    $month = substr($date_ar[0], 2, 2);
    $date = substr($date_ar[0], -2);
    switch ( $month ) {
        case "1":
            $monthstring = "січня";
            break;
        case "2":
            $monthstring = "лютого";
            break;
        case "3":
            $monthstring = "березня";
            break;
        case "4":
            $monthstring = "квітня";
            break;
        case "5":
            $monthstring = "травня";
            break;
        case "6":
            $monthstring = "червня";
            break;
        case "7":
            $monthstring = "липня";
            break;
        case "8":
            $monthstring = "серпня";
            break;
        case "9":
            $monthstring = "вересня";
            break;
        case "10":
            $monthstring = "жовтня";
            break;
        case "11":
            $monthstring = "листопаду";
            break;
        case "12":
            $monthstring = "грудня";
            break;
        default:
            $monthstring = "-1";
    }
}
```

```

$newdate = $date." ".$monthstring." 20".$year." ".$time;
return $newdate; // повернення коду дати
}

```

```

// Функція конвертування дати
function ConvertDateDB($DateDB)
{

```

```

$year = substr($DateDB,0,2);
$month = substr($DateDB,-2);
    switch ( $month ) {
        case "1":
            $monthstring = "січень";
            break;
        case "2":
            $monthstring = "лютий";
            break;
        case "3":
            $monthstring = "березень";
            break;
        case "4":
            $monthstring = "квітень";
            break;
        case "5":
            $monthstring = "травень";
            break;
        case "6":
            $monthstring = "червень";
            break;
        case "7":
            $monthstring = "липень";
            break;
        case "8":
            $monthstring = "серпень";
            break;
        case "9":
            $monthstring = "вересень";
            break;
        case "10":
            $monthstring = "жовтень";
            break;
        case "11":
            $monthstring = "листопад";
            break;
        case "12":
            $monthstring = "грудень";
            break;
        default:
            $monthstring = "-1";
    }

```

```

$newdate = $monthstring." 20".$year;
return $newdate; // повернення коду дати
}

```

```

// Функція отримання поточного місяця
function GetMonths()
{

```

```

    global $link;
    $query = "SELECT yearmonth FROM counter WHERE `ip`=1";
    if (!( $result = mysql_query($query, $link) ))
    {
        echo ( "Помилка при виконанні $query
".mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
//друкування результату
    $i = 1;

```

```

while (( $row = mysql_fetch_object( $result ))
{
    $month[$i] = $row -> yearmonth;
    $i++;
}
mysql_free_result( $result );
return $month; // повернення коду місяця
}

// Функція лічильника
function ShowCounter($ip, $dateDB)
{
    global $link;
    $query = "SELECT ip,counter, yearmonth FROM counter WHERE `ip`=$ip AND
`yearmonth`=$dateDB";
    if (!( $result = mysql_query($query, $link ) ))
    {
        echo ( "Помилка при виконанні $query
".mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
    //друкування результату
    echo("<table cellpadding=0 cellspacing=5><tr><td colspan=2
class=left>");
    echo("</td></tr>");
    while (( $row = mysql_fetch_object( $result ))
    {
        printf("<tr><td>%s</td><td>%s</td><td>%s</td></tr>\n",
$row -> yearmonth, $row -> ip, $row -> counter);
    }
    echo("</table>");
    mysql_free_result( $result );
}

// Функція роботи з трафіком
function TrafRemainder($balance, $group, $counter, $moneymonth)
{
    global $link;
    $query_group = "SELECT `trafcost`,`abontraf` FROM `group` WHERE
`group`='".$group."'";
    if (!( $result_group = mysql_query($query_group, $link ) ))
    {
        echo ( "Помилка при виконанні $query
".mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
    $row_group = mysql_fetch_object( $result_group );
    if ($row_group -> trafcost > 0)
    {
        if ($moneymonth=="y")
        {
            if ($counter < ($row_group -> abontraf*1048576))
            {
                $TrafRemainder = number_format(($row_group -> abontraf*1048576-
$counter)/1048576+ConvertMoneyFromDB(($balance)/(($row_group -> trafcost)/100),
2, chr(44), "."));
            } else {
                $TrafRemainder = ConvertMoneyFromDB(($balance)/(($row_group ->
trafcost)/100));
            }
        } else { $TrafRemainder = 0; }
    } else {
        $TrafRemainder = "-";
    }
}
return $TrafRemainder; // повернення коду трафіку
}

// Функція відображення сумарної статистики
function ShowStat()

```

```

{
    global $link, $monthtoshow;
$query1 = "SELECT `ip` FROM `user` WHERE `access`='y'";
$query2 = "SELECT `ip` FROM `user`;
if ($monthtoshow)
{
$query3 = "SELECT `money` FROM `money` WHERE `date` LIKE '". $monthtoshow. "%'";
$query4 = "SELECT `counter` FROM `counter` WHERE `yearmonth` LIKE
'". $monthtoshow. "'";
} else {
$query3 = "SELECT `money` FROM `money` WHERE `date` LIKE
'".GetCurrentDateDB()."%'";
$query4 = "SELECT `counter` FROM `counter` WHERE `yearmonth` LIKE
'".GetCurrentDateDB()."';
}

    if (!( $result1 = mysql_query($query1, $link ) ))
    {
        echo ( "Помилка при виконанні $query1
".mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
    if (!( $result2 = mysql_query($query2, $link ) ))
    {
        echo ( "Помилка при виконанні $query2
".mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
    if (!( $result3 = mysql_query($query3, $link ) ))
    {
        echo ( "Помилка при виконанні $query3
".mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
    if (!( $result4 = mysql_query($query4, $link ) ))
    {
        echo ( "Помилка при виконанні $query4
".mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
    $activeusers=mysql_num_rows($result1);
    $allusers=mysql_num_rows($result2);
    while ( $row_money = mysql_fetch_object( $result3 ) )
    {
        $allmoney+=$row_money -> money;
    }
    while ( $row_counter = mysql_fetch_object( $result4 ) )
    {
        $alltraffic+=$row_counter -> counter;
    }
    while ( $row_user = mysql_fetch_object( $result2 ) )
    {
        if ($monthtoshow)
        {
            $query5 = "SELECT `counter` FROM `counter` WHERE `ip`='". $row_user -> ip.'" AND
`yearmonth` LIKE '". $monthtoshow. "'";
        } else {
            $query5 = "SELECT `counter` FROM `counter` WHERE `ip`='". $row_user -> ip.'"
AND `yearmonth` LIKE '".GetCurrentDateDB()."';
        }
        if (!( $result5 = mysql_query($query5, $link ) ))
        {
            echo ( "Помилка при виконанні $query5
".mysql_errno($link)." ".mysql_error($link) );
            exit();
        }
        $row_counterusers = mysql_fetch_object( $result5 );
        $alltrafficusers+=$row_counterusers -> counter;
    }
?>

```

```

<table cellpadding="0" cellspacing="0" border="0">
<tr class="odd"><td>
<a href="<? echo ("$_PHP_SELF"."?action=");?>">Вернуться</a>
</td></tr>
<tr class="odd"><td valign="top">
<table cellpadding="0" cellspacing="0" border="0">
<tr class="edge"><td>
    <table cellpadding="3" cellspacing="1" border="0">
    <tr class="odd">
    <td height=30>Кількість активних користувачів</td><td width="30"
align="center"><? echo $activeusers;?></td>
    </tr>
    <tr class="odd">
    <td height=30>Загальна кількість користувачів</td><td width="30"
align="center"><? echo $allusers;?></td>
    </tr>
    <tr class="odd">
    <td height=30>Кількість зібраних грошей (грн.)</td><td width="30"
align="center"><? echo number_format($allmoney,0, chr(44), ".");?></td>
    </tr>
    <tr class="odd">
    <td height=30>Загальний трафік:</td><td width="30" align="center"><? echo
number_format($alltraffic,0, chr(44), ".");?></td>
    </tr>
    <tr class="odd">
    <td height=30>Загальний трафік (використовуючи таблицю user):</td><td
width="30" align="center"><? echo number_format($alltrafficusers,0, chr(44),
".");?></td>
    </tr>
</table>
</td>
</tr>
<tr class="odd"><td>
    <form method="GET" action="<? echo ("$_PHP_SELF");?>">
    <select name="monthtoshow">
<?
    $month = GetMonths();
    $i=1;
    while ($month[$i])
    {
    if (GetCurrentDateDB()==$month[$i])
    {
    echo "<OPTION VALUE=\"\$month[$i]\"
SELECTED>".ConvertDateDB($month[$i])."</option>";
    } else {
    echo "<OPTION
VALUE=\"\$month[$i]\">".ConvertDateDB($month[$i])."</option>";
    }
    $i++;
    }
?>
    </select>
    <input type="hidden" name="action" value="stat">
    <br>
    <input type="submit" value="Показать">
    </form>
</td></tr>
</table>
<?
}

// Функція оновлення групи
function UpdateGroup()
{
    global $link, $group, $abon, $abontraf, $trafcost, $message;
    $query = "UPDATE `group` SET `abon`='". $abon.'" ,
`abontraf`='". $abontraf.'" , `trafcost`='". $trafcost.'" WHERE
`group`='". $group.'" ";
    if (!( $result = mysql_query($query, $link ) ))
    {

```

```

        echo ( "Помилка при виконанні $query ".mysql_errno($link)." ".
            mysql_error($link) );
            exit();
        } else {
            $message="Updated";
        }
    }

// Функція видалення групи
function DelGroup()
{
    global $link, $group, $message;
    $query = "DELETE FROM `group` WHERE `group`='".$group."'";
    if (!( $result = mysql_query($query, $link) ))
    {
        echo ( "Помилка при виконанні $query
".mysql_errno($link)." ".mysql_error($link) );
        exit();
    } else {
        $message="Група ".$group." удалена";
    }
}

// Функція додавання групи
function AddGroup()
{
    global $link, $groupname;
    if ($groupname)
    {
        $query_add = "INSERT INTO `group` ( `group` ) VALUES ( '$groupname' )";
        if (!( $result = mysql_query($query_add, $link) ))
        {
            echo ( "Помилка при виконанні $query_add
".mysql_errno($link)." ".mysql_error($link) );
            exit();
        }
    }
}

// Функція редагування групи
function EditGroup()
{
    global $link, $group, $message;
    ?>
<table cellpadding="0" cellspacing="0" border="0">
<tr class="odd"><td>
<a href="<? echo ("$_PHP_SELF"."?action=");?>">Вернуться</a>
</td></tr>
<tr class="odd"><td valign="top">
<table cellpadding="0" cellspacing="0" border="0">
<tr class="edge"><td>
<table cellpadding="3" cellspacing="1" border="0">
<form method="POST" action="<? echo ("$_PHP_SELF");?>">
<tr class="even"><td height=30>
    Група:
</td>
<td bgcolor="#ffffff" width="110">
<select name="group">
<?
    $query_group = "SELECT `group` FROM `group`";
    if (!( $result_group = mysql_query($query_group, $link) ))
    {
        echo ( "Помилка при виконанні $query_group
".mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
    while ( $row_group = mysql_fetch_object( $result_group ) )
    {
        if ($row_group -> group == $group)

```

```

        {
        ?>
        <OPTION VALUE="<? echo $row_group -> group; ?>" SELECTED><? echo
$row_group -> group; ?></option>";
        <?
        } else {
        ?>
        <OPTION VALUE="<? echo $row_group -> group; ?>"><? echo
$row_group -> group; ?></option>";
        <?
        }
        }
?>

</select>
</td></tr>
<tr class="odd"><td></td><td height=30 align="left">
<input type="submit" value="Показати">
<input type="hidden" name="action" value="editgroup" >
</form>
</td></tr>
<tr class="even"><td colspan="2">
Додати групу:
</td></tr>
<tr class="odd"><td>
<form method="POST" action="<? echo ("$_PHP_SELF");?>">
<input type="text" size=8 name="groupname">
</td><td>
<input type="hidden" name="action" value="addgroup" >
<input type="submit" value="Додати">
</form>
</td></tr>
</table>
</td></tr>
<? if ($group)
{?>
<tr class="odd"><td>
<a href="<?echo ("$_PHP_SELF"?action=delgroup&group=$group)"?>"
onclick="return confirmLink(this, 'видалити групу `<? echo
$group;?>`')">Видалити групу <?echo $group;?></a>
</td></tr>
<?}?>
</table>
</td>

<td width="500" align="right" valign="top">
<table cellpadding="0" cellspacing="0" border="0">
<tr><td>
<?
if ($group != "")
{
$query_group = "SELECT `trafcost`, `abon`, `abontraf` FROM `group` WHERE
`group`='<?echo $group;?>'";
if (!( $result_group = mysql_query($query_group, $link ) ) )
{
echo ( "Помилка при виконанні $query_group
".mysql_errno($link)." ".mysql_error($link) );
exit();
}
$row_group = mysql_fetch_object( $result_group );
?>

<table cellpadding="0" cellspacing="0" border="0">
<tr class="edge"><td>
<form name="updategroup" method="POST" action="<?echo
("$_PHP_SELF");?>>
<table cellpadding="2" cellspacing="1" border="0">
<tr class="odd"><td>Вартість трафіка (грн. за 100
Мбайт):</td><td align="right"><input type="text" name="trafcost" value="<? echo
$row_group -> trafcost; ?>" size="2"></td></tr>

```

```

                <tr class="odd"><td>Абонентська плата (грн. за
місяць):</td><td align="right"><input type="text" name="abon" value="<? echo
$row_group -> abon; ?>" size="2"></td></tr>
                <tr class="odd"><td>Кількість трафіка, вхідного в
абонентську плату (Мбайт):</td><td align="right"><input type="text"
name="abontraf" value="<? echo $row_group -> abontraf; ?>" size="2"></td></tr>
                <tr class="odd"><td colspan="2" align="right"><input
type="submit" value="Відіслати"></td></tr>
            </table>
        </td></tr>
        <input type="hidden" name="group" value="<? echo $group; ?>">
        <input type="hidden" name="action" value="updategroup">
    </form>
</table>
<?
}
?>
</td></tr>
</table>
</td>
<td valign="top" align="right" width="70"><font color="red"><? echo
$message;?></td>
</tr>
</table>
<?
}

// Функція додавання користувача
function AddUser()
{
    global $link, $login, $ip, $name, $address, $moneyinet, $moneynetwork,
$balance, $group, $trafficebegin, $SubtractAb;
    $balanceDB = ConvertMoneyToDB($balance);
    $query_user = "INSERT INTO `user` ( `login`, `ip`, `name`, `address`,
`moneyinet`, `moneynetwork`, `balance`, `group`) VALUES ( '$login', '$ip',
'$name', '$address', '$moneyinet', '$moneynetwork', '$balanceDB', '$group' )";
    if (!( $result = mysql_query($query_user, $link ) ) )
    {
        echo ( "Помилка при виконанні $query_user
".mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
    if ($trafficebegin)
    {
        $query_counter = "UPDATE `counter` SET `counter`='$trafficebegin',
`moneymonth`='$SubtractAb' WHERE `ip`='$ip' AND
`yearmonth`='".GetCurrentDateDB()."";
    } else {
        $query_counter = "UPDATE `counter` SET `counter`='0',
`moneymonth`='$SubtractAb' WHERE `ip`='$ip' AND
`yearmonth`='".GetCurrentDateDB()."";
    }
    if (!( $result_counter = mysql_query($query_counter, $link ) ) )
    {
        echo ( "Помилка при виконанні $query_counter
".mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
    $date = GetCurrentDateDBMoney();
    $query_money = "INSERT INTO `money` ( `ip`, `money`, `date`) VALUES (
'$ip', '$balance', '$date' )";
    if (!( $result = mysql_query($query_money, $link ) ) )
    {
        echo ( "Помилка при виконанні $query_money
".mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
}
}

```

```

// Функція форми користувача
function AddUserForm()
{
    global $link;
?>
<table cellpadding="0" cellspacing="0" border="0">
<tr class="odd"><td>
<a href="<? echo ("$_PHP_SELF"."?action=");?>">Повернутися</a>
</td></tr>
<tr><td>
<table cellpadding="0" cellspacing="0" border="0">
<tr class="edge"><td>
    <table cellpadding="3" cellspacing="1" border="0">
    <form method="GET" action=<?echo ("$_PHP_SELF");?>>
    <tr class="even"><td height=30>
        Login:
    </td>
    <td bgcolor="#ffffff">
        <input type="text" name="login" size="12">
    </td>
    <td bgcolor="#ffffff">
        маленькими літерами
    </td>
    </tr>
<tr class="even"><td height=30>
        IP:
    </td>
    <td bgcolor="#ffffff">
        <input type="text" name="ip" size="12">
    </td>
    <td bgcolor="#ffffff">
        уводиться тільки остання частина ip-адреса
    </td>
    </tr>
<tr class="even"><td height=30>
        Имя:
    </td>
    <td bgcolor="#ffffff">
        <input type="text" name="name" size="12">
    </td>
    <td bgcolor="#ffffff">
        уводиться у вигляді: Фамилія Ім'я
    </td>
    </tr>
<tr class="even"><td height=30>
        Адреса:
    </td>
    <td bgcolor="#ffffff">
        <input type="text" name="address" size="12">
    </td>
    <td bgcolor="#ffffff">
        уводиться в вигляді: 15/1 7-2-3
    </td>
    </tr>
<tr class="even"><td height=30>
        Деньги за сеть:
    </td>
    <td bgcolor="#ffffff">
        <input type="text" name="moneynetwork" size="12">
    </td>
    <td bgcolor="#ffffff"></td>
    </tr>
<tr class="even"><td height=30>
        Деньги за инет:
    </td>
    <td bgcolor="#ffffff">
        <input type="text" name="moneyinet" size="12">
    </td>
    <td bgcolor="#ffffff"></td>

```

```

</tr>
<tr class="even"><td height=30>
    Баланс:
</td>
<td bgcolor="#ffffff">
    <input type="text" name="balance" size="12">
</td>
<td bgcolor="#ffffff"></td>
</tr>
<tr class="even"><td height=30>
    Вилучити абонентську плату?
</td>
<td bgcolor="#ffffff">
    <input type="radio" name="SubtractAb" value="n" checked>Да&nbsp;nbsp;
    <input type="radio" name="SubtractAb" value="y">Нет&nbsp;nbsp;
</td>
<td bgcolor="#ffffff"></td>
</tr>
<tr class="even"><td height=30>
    Початковий трафік (байт):
</td>
<td bgcolor="#ffffff">
    <input type="text" name="trafficbegin" value="0" size="12">
</td>
<td bgcolor="#ffffff"></td>
</tr>
<tr class="even"><td height=30>
    Група:
</td>
<td bgcolor="#ffffff">
    <select name="group">
<?
    $query_group = "SELECT `group` FROM `group`";
    if (!( $result_group = mysql_query($query_group, $link) ))
    {
        echo ( "Помилка при виконанні $query_group
".mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
    while ( $row_group = mysql_fetch_object( $result_group ) )
    {
        if ($row_group -> group=="comp")
        {
            ?>
            <OPTION VALUE="<? echo $row_group -> group; ?>" SELECTED><? echo
$row_group -> group; ?></option>";
            <?
        } else {
            ?>
            <OPTION VALUE="<? echo $row_group -> group; ?>"><? echo
$row_group -> group; ?></option>";
            <?
        }
    }
?>
    </select>
</td>
<td bgcolor="#ffffff"></td>
</tr>
<tr class="odd"><td></td><td height=30 align="left">
    <input type="submit" value="Додати">
</td>
<td bgcolor="#ffffff"></td>
</tr>
    <input type="hidden" name="action" value="adduser" >
</form>
</table>
</td></tr>
</table>

```

```

</td>
</tr>
</table>
<?
}

// Функція видалення користувача
function DeleteUser ()
{
    global $link, $ip;
    $query = "DELETE FROM `user` WHERE `ip`='". $ip. "'";
    if (!( $result = mysql_query($query, $link) ))
    {
        echo ( "Помилка при виконанні $query ".mysql_errno($link)."
            ".mysql_error($link) );
        exit();
    }
    $query_counter = "UPDATE `counter` SET `counter`='0', `moneymonth`='n'
WHERE `ip`='$ip' AND `yearmonth`='".GetCurrentDateDB()." ";
    if (!( $result_counter = mysql_query($query_counter, $link) ))
    {
        echo ( "Помилка при виконанні $query_counter
".mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
}

// Функція оновлення даних користувача
function UpdateUser ()
{
    global $link, $message, $ip, $login, $group, $address, $moneynetwork,
$moneynet;
    $query = "UPDATE `user` SET `login`='". $login. "', `group`='". $group. "',
`address`='". $address. "', `moneynetwork`='". $moneynetwork. "',
`moneynet`='". $moneynet. "' WHERE `ip`='". $ip;
    if (!( $result = mysql_query($query, $link) ))
    {
        echo ( "Помилка при виконанні $query
".mysql_errno($link)." ".mysql_error($link) );
        exit();
    } else {
        $message = "Updated";
    }
}

// Функція додавання до балансу
function IncBalance ()
{
    global $link, $message, $ip, $incbalance ;
    if ($incbalance)
    {
        $query = "UPDATE `user` SET balance = balance +
".ConvertMoneyToDB($incbalance)." WHERE `ip`='". $ip;
        if (!( $result = mysql_query($query, $link) ))
        {
            echo ( "Помилка при виконанні $query ".mysql_errno($link)."
            ".mysql_error($link) );
            exit();
        }
        $date = GetCurrentDateDBMoney();
        $query_money = "INSERT INTO `money` ( `ip`, `money`, `date`) VALUES (
            '$ip', '$incbalance', '$date' )";
        if (!( $result = mysql_query($query_money, $link) ))
        {
            echo ( "Помилка при виконанні $query_money
            ".mysql_errno($link)." ".mysql_error($link) );
            exit();
        } else {
            $message = "Updated";
        }
    }
}

```

```

    }
}

// Функція відображення даних адміністратора
function ShowUserAdmin($ip, $month)
{
    global $link, $message, $showpayment;
    $query_user = "SELECT `name`, `login`, `access`, `balance`, `group`, `address`,
        `moneynetwork`, `moneyinet` FROM user WHERE `ip`=$ip";
    if (!( $result_user = mysql_query($query_user, $link) ))
    {
        echo ( "Помилка при виконанні $query_user
            ".mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
    $row_user = mysql_fetch_object( $result_user );
    $query_group = "SELECT `group` FROM `group`";
    if (!( $result_group = mysql_query($query_group, $link) ))
    {
        echo ( "Помилка при виконанні $query_group ".mysql_errno($link)."
            ".mysql_error($link) );
        exit();
    }
    ?>
<table cellpadding="0" cellspacing="0" border="0">
<tr class="odd"><td>
<a href="<? echo (" $PHP_SELF"."?action=");?>">Повернутися</a>
</td></tr>
<tr class="odd" valign="top"><td>
<table cellpadding="0" cellspacing="0" border="0">
<tr class="edge"><td>
<form name="update" method="POST" action=<?echo (" $PHP_SELF");?>>
<table cellpadding="2" cellspacing="1" border="0">
<?
                $query_counter = "SELECT counter, moneymonth FROM
counter WHERE `ip`=". $ip ." AND `yearmonth`=". $month;
                if (!( $result_counter = mysql_query($query_counter, $link) ))
                {
                    echo ( "Помилка при виконанні $query_counter
                        ".mysql_errno($link)." ".mysql_error($link) );
                    exit();
                }
                $row_counter = mysql_fetch_object( $result_counter );
    ?>
<tr class="odd"><td>Дані за:</td><td align="right"><? echo
ConvertDateDB($month);?></td></tr>
<tr class="even"><td>IP:</td><td align="right"><? echo $ip;?></td></tr>
<tr class="odd"><td>Login:</td><td align="right"><input type="text" name="login"
value="<? echo $row_user -> login; ?>" size="8"></td></tr>
<tr class="even"><td>Група:</td><td align="right">
                <select name="group">
<?
                while ( $row_group = mysql_fetch_object( $result_group ) )
                {
                    if ($row_group -> group==$row_user -> group)
                    {
                        ?>
                        <OPTION VALUE="<? echo $row_group -> group; ?>" SELECTED><? echo
$row_group -> group; ?></option>";
                        <?
                    } else {
                        ?>
                        <OPTION VALUE="<? echo $row_group -> group; ?>"><? echo
$row_group -> group; ?></option>";
                        <?
                    }
                }
            ?>

```

```

    }
    }
?>
    </select>
        </td></tr>
        <tr class="odd"><td>Имя:</td><td align="right"><? echo
$row_user -> name; ?></td></tr>
        <tr class="even"><td>Адрес:</td><td align="right"><input
type="text" name="address" value="<? echo $row_user -> address; ?>"
size="8"></td></tr>
        <tr class="odd"><td>Гроші за мережу:</td><td
align="right"><input type="text" name="moneynetwork" value="<? echo $row_user ->
moneynetwork; ?>" size="2"></td></tr>
        <tr class="even"><td>Гроші за инет:</td><td
align="right"><input type="text" name="moneynet" value="<? echo $row_user ->
moneynet; ?>" size="2"></td></tr>
        <tr class="odd"><td>Трафік (байт):</td><td
align="right"><? echo number_format($row_counter -> counter,0, chr(44), ".");
?></td></tr>
        <tr class="even"><td>Баланс (грн.):</td><td
align="right"><font color="red"><? echo ConvertMoneyFromDB($row_user ->
balance); ?></font></td></tr>
        <tr class="odd"><td>balance:</td><td align="right"><?
echo $row_user -> balance; ?></td></tr>
        <tr class="even"><td>Залишилося (МБ):</td><td
align="right"><? echo TrafRemainder($row_user -> balance, $row_user -> group,
$row_counter -> counter, $row_counter -> moneymonth); ?></td></tr>
        <tr class="odd"><td>Доступ:</td><td align="right"><?
echo $row_user -> access; ?></td></tr>
        <tr class="even"><td>Абон. плата:</td><td
align="right"><? echo $row_counter -> moneymonth; ?></td></tr>
        <tr class="odd"><td colspan=2 align="center"><input
type="submit" value="Відправити"></td></tr>
        <input type=hidden name="ip" value="<? echo $ip;?>" >
        <input type=hidden name="month" value="<? echo $month;?>" >
        <input type=hidden name="action" value="update" >
        </form>
    </table>
</td></tr>
</table>
</td>
    <td></td>
    <td valign="top" align="right" width="300">
    <table cellpadding="0" cellspacing="0" border="0">
    <tr class="edge"><td>
        <table cellpadding="2" cellspacing="1" border="0">
        <form name="incbalance" method="POST" action=<?echo ("$_PHP_SELF");?>>
        <tr class="odd">
            <td>Баланс (грн.):</td>
            <td><font color="red"><? echo ConvertMoneyFromDB($row_user -> balance);
?></font></td>
        </tr>
        <tr class="odd">
            <td>Додати (грн.):</td>
            <td><input type="text" name="incbalance" size="8"></td>
        </tr>
        <tr><td bgcolor="#ffffff" height=40 colspan=2 align="center">
            <input type=hidden name="ip" value="<? echo $ip;?>" >
            <input type=hidden name="month" value="<? echo $month;?>" >
            <input type=hidden name="action" value="incbalance" >
            <input type="submit" value="Відправити">
        </td></tr>
        </form>
    </table>
</td></tr>
</table>

<table>
    <tr class="odd" colspan="2">

```

```

<td colspan="2" align="right">
<?
if ($showpayment)
{?>
<a href="<? echo (" $PHP_SELF"."?action=show&ip=$ip&month=$month");?>">Вилучити
платежі</a>
</td></tr>
</table>
<table cellpadding="0" cellspacing="0" border="0">
<tr class="edge"><td>
<table cellpadding="2" cellspacing="1" border="0">
<tr class="odd"><td>Дата</td><td>руб.</td></tr>
<?
$query_money = "SELECT money, date FROM money WHERE `ip`=".$ip." AND `date` like
'".$month."%";
if (!( $result_money = mysql_query($query_money, $link ) ))
{
echo ("Помилка при виконанні $query_money ".mysql_errno($link)."
.mysql_error($link) );
exit();
}
while ($row_money = mysql_fetch_object( $result_money ))
{
echo "<tr class=\"odd\"><td>".ConvertDateDBMoney($row_money ->
date)."</td><td>".$row_money -> money."</td></tr>";
$allmoney+=$row_money -> money;
}
echo "<tr class=\"odd\"><td>Итого:</td><td>".$allmoney."</td></tr>";
echo "</table>";
} else {?>
<a href="<? echo
("$PHP_SELF"."?action=show&ip=$ip&month=$month&showpayment=1");?>">
Дивитися платежі</a>
<?>?>
</td></tr>
</table>
</td>
<td valign="top"><font color="red"><? echo $message;?></td>
</tr>
</table>
<?
}

// Функція відображення користувача
function ShowUser($ip)
{
global $link;
$query_user = "SELECT `name`, `login`, `access`, `balance`, `group` FROM
user WHERE `ip`=$ip";
if (!( $result_user = mysql_query($query_user, $link ) ))
{
echo ( "Помилка при виконанні $query ".mysql_errno($link)."
.mysql_error($link) );
exit();
}
$row_user = mysql_fetch_object( $result_user );
//друкування результату
?>

<table cellpadding="0" cellspacing="0" border="0">
<tr class="edge"><td>
<table cellpadding="2" cellspacing="1" border="0">
<?
$query_counter = "SELECT counter, moneymonth FROM
counter WHERE `ip`=".$ip." AND `yearmonth`=".GetCurrentDateDB();
if (!( $result_counter =
mysql_query($query_counter, $link ) ))
{

```

```

echo ( "Помилка при виконанні
$query_counter ".mysql_errno($link)." ".mysql_error($link) );
exit();
}
$row_counter = mysql_fetch_object(
$result_counter );
// виведення даних на екран
printf("
<tr class=\"odd\"><td>Дані за:</td><td align=\"right\">%s</td></tr>
<tr class=\"even\"><td>Login:</td><td align=\"right\">%s</td></tr>
<tr class=\"odd\"><td>Ім'я:</td><td align=\"right\">%s</td></tr>
<tr class=\"even\"><td>Трафік (байт):</td><td align=\"right\">%s</td></tr>
<tr class=\"odd\"><td>Баланс (грн.):</td><td align=\"right\"><font
color=\"red\">%s</font></td></tr>
<tr class=\"even\"><td>Залишилося (Мб):</td><td align=\"right\">%s</td></tr>
<tr class=\"odd\"><td>Доступ:</td><td align=\"right\">%s</td></tr>
<tr class=\"even\"><td>Абон. плата:</td><td align=\"right\">%s</td></tr>
\n",
ConvertDateDB(GetCurrentDateDB()), $row_user -> login, $row_user -> name,
number_format($row_counter -> counter,0, chr(44), "."),
ConvertMoneyFromDB($row_user -> balance), TrafRemainder($row_user -> balance,
$row_user -> group, $row_counter -> counter, $row_counter -> moneymonth),
$row_user -> access, $row_counter -> moneymonth);
?>
</table>
</td></tr>
</table>
<?
}

// Функція попередження
function ResolveUser()
{
global $network, $conf;
$host = getenv('REMOTE_ADDR');
$ip=explode(".", $host);
$network = $ip[0].".$ip[1].".$ip[2].".";
if ($network != $conf{'network'})
{
echo ("Зареєструйтеся в VPN.");
exit();
}
return $ip[3]; // повернення коду IP
}

// Функція роботи з датою
function GetCurrentDateDB()
{
$current_date=time();
$dateDB=date("ym", $current_date);
return $dateDB; // повернення коду дати у БД
}

function GetCurrentDateDBMoney()
{
$current_date=time();
$date=date("ymd H:i:s", $current_date);
return $date; // повернення коду дати внесення коштів
}

function Stuff()
{
$current_date=time();
echo $current_date."<br>";
$date=date("ymd H:i:s", $current_date);
$dateDB=date("ym", $current_date);
echo $date."<br>".$dateDB."<br>";
echo ConvertDateDBMoney($date);
}

```

```
// функція формування початку HTML сторінки
function Top($path,$pagename)
{
?>
<html>
<head>
<title>Моніторинг трафіку -> <? echo $pagename ?></title>
<META HTTP-EQUIV="Expires" CONTENT="Fri, Jan 01 1900 00:00:00 GMT">
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=windows-1251">
<link rel="stylesheet" type="text/css" href="<? echo $path; ?>Monitoring.css">

<script type="text/javascript" language="javascript">
<!--
// коди та текст помилок
var errorMsg0 = 'Потрібно значення для форми!';
var errorMsg1 = 'Це не число!';
var errorMsg2 = 'Неприпустима кількість рядків!';
var noDropDbMsg = 'Команда "Видалити БД" відключена.';
var confirmMsg = 'Ви дійсно бажаєте ';
//-->
</script>
<script src="<? echo $path;?>functions.js" type="text/javascript"
language="javascript"></script>
</head>
<body>
<?
}

// формування HTML коду кінця сторінки
function Footer()
{
?>
</body>
</html>
<?
}
?>
```