

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 першим (бакалаврським) рівнем вищої освіти**  
на тему  
**“ Програмне забезпечення підсистеми системного**  
**адміністрування та управління неоднорідною мережею”**

Виконав здобувач вищої освіти  
IV курсу, групи КМ-19  
ОПП «Комп’ютерна інженерія»  
спеціальності 123 «Комп’ютерна інженерія»  
\_\_\_\_\_ Козак А.І.  
« \_\_\_\_ » \_\_\_\_\_ 2023 р.

Керівник проекту  
кандидат технічних наук, доцент  
\_\_\_\_\_ Пархоменко Ю.М.  
« \_\_\_\_ » \_\_\_\_\_ 2023 р.  
Рецензент \_\_\_\_\_

м. Кропивницький

# Центральноукраїнський національний технічний університет

Факультет Механіко-технологічний

Кафедра Кібербезпеки та програмного забезпечення

Рівень вищої освіти бакалавр

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

" " 2023 року

## ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ПЕРШИМ (БАКАЛАВРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

*Козаку Артему Ігоровичу*

(прізвище, ім'я, по батькові)

1. Тема роботи Програмне забезпечення підсистеми системного адміністрування та управління неоднорідною мережею

2. Керівник роботи Пархоменко Юрій Михайлович, канд. техн. наук, доцент  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від " 5 " січня 2023 року № 10-02

3. Строк подання роботи до захисту 22.05.2023 р.

4. Мета та завдання випускної кваліфікаційної роботи. *Метою роботи є розробка програмного забезпечення підсистеми системного адміністрування та управління неоднорідною мережею*

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

*1. Призначення та область використання.*

*2. Перегляд аналогічних існуючих систем.*

*3. Опис і обґрунтування проектних рішень.*

*4. Етапи програмування системи.*

*5. Впровадження системи в промислову експлуатацію*

*6. Висновки.*

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

*Структурна схема системи* 1 аркуш

*Функціональна схема системи* 1 аркуш

*Діаграма процесів* 1 аркуш

*Блок-схема алгоритму роботи додатку* 2 аркуша

7. Дата видачі завдання «  »    20   р.

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.03.2023 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2023 р.	
3.	Розробка моделі компонента	20.03.2023 р.	
4.	Розробка структур даних	25.03.2023 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.03.2023 р.	
6.	Програмування алгоритмів	10.04.2023 р.	
7.	Оформлення ПЗ	17.05.2023 р.	
8.	Попередній захист роботи	22.05.2023 р.	

Дата видачі завдання  
«  »    2023р.

Підпис керівника

\_\_\_\_\_ (прізвище та ініціали)

Завдання прийнято до виконання  
«  »    2023р.

Підпис здобувача

\_\_\_\_\_ (прізвище та ініціали)

## АНОТАЦІЯ

**Козак А.І. Програмне забезпечення підсистеми системного адміністрування та управління неоднорідною мережею. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2023.**

В даній бакалаврській дипломній роботі розроблена система маршрутизації на базі SoftLinux рішення, яка призначена для об'єднання різномірних, віддалених мереж у єдину систему.

В процесі роботи над проектом виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленої системи. Застосовані правила маршрутизації з урахуванням одночасного використання декількох каналів Інтернет. Система забезпечує надійність та захист всіх компонентів мережі.

Система розгорнута на ПЕОМ архітектури IBM PC з ОС Linux Ubuntu server 20. Було використано програмне забезпечення Iptables, dnsmasq та вбудовані модулі ядра ОС.

**Ключові слова:** комп'ютерна інженерія, маршрутизація, програмно-апаратні рішення, управління, системне адміністрування, неоднорідні мережі.

## ABSTRACT

**Kozak A.I. Subsystem software for system administration and heterogeneous network management. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2023**

In this bachelor's thesis, a routing system based on SoftLinux solution is developed, which is designed to combine heterogeneous, remote networks into a single system.

In the process of working on the project, an analysis of existing hardware and software was performed. All components of the developed system are fully described. Routing rules are applied taking into account the simultaneous use of several Internet channels. The system provides reliability and protection of all network components.

The system is deployed on a PC IBM PC with Linux Ubuntu server 20. Iptables, dnsmasq software and OS kernel modules were used.

**Keywords:** computer engineering, routing, software and hardware solutions, management, system administration, heterogeneous networks.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ .....	3
ВСТУП .....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ .....	7
1.1 Призначення системи.....	7
1.2 Область застосування .....	11
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ.....	14
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським рівнем вищої освіти) .....	14
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування .....	22
2.3 Розгорнута постановка завдання .....	23
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ .....	25
3.1 Опис функціонування системи .....	25
3.2 Розробка структурної схеми.....	32
3.3. Розробка функціональної схеми .....	34
3.4. Розробка діаграми процесів. ....	35
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ ...	38
4.1 Розробка блок-схем та опис алгоритмів функціонування системи ...	38
4.4 Захист розробленого програмного забезпечення.....	53

						ВКРБ-123.23.0004.00.00.ПЗ		
Ви	Арк.	№ докум.	Підп.	Дата				
Розроб.		Козак А.І.			Програмне забезпечення підсистеми системного адміністрування та управління неоднорідною мережею	Літ.	Аркуш	Аркушів
Перев.		Пархоменко Ю.М.				Б	1	75
Н.контр.р		Гермак В.С.			ЦНТУ КМ-19			
Затв.		Смірнов О.А.						

5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ.....	57
6 ОСНОВНІ ВИСНОВКИ.....	61
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ7 .....	64

Кафедра \_ КБПЗ \_ 2023рік

					<b>ВКРБ-123.23.0004.00.00.ПЗ</b>	Арк.
<i>Вим.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		2



## ВСТУП

Управління комп'ютерними мережами ніколи не займало домінуючого положення в комп'ютерних технологіях. Відводилася їм традиційно незначна роль. Це призвело до того, що структура і функціональні можливості цього класу програмного забезпечення (ПЗ) безпосередньо залежали від архітектури комп'ютерних систем і зростали разом з ними.

Швидке розширення розподіленої клієнт-серверної архітектури призвело до радикальних змін у сфері управління інформаційними системами. Різні ресурси: різні комп'ютерні платформи, активне мережеве апаратне та програмне забезпечення, необхідні для вирішення абсолютно нових адміністративних завдань: управління розподіленими ресурсами, електронне розповсюдження програмного забезпечення, аналіз трафіку та керування пропускнуою здатністю мережі, перерозподіл серверів завантаження, моніторинг стану індивідуальної настільної системи тощо. Однак адміністративні додатки, що працюють на мейнфреймах, не можна було перенести в нове середовище, тому виробникам довелося створювати програмне забезпечення для керування майже з нуля.

Поява розподіленої архітектури змінила принципи управління. Спочатку завдання управління обмежувалося моніторингом продуктивності окремих компонентів (мережеві пристрої, ПК і робочі станції, пристрої зберігання даних, периферійні пристрої тощо), і в багатьох випадках це було просто питання збору даних про ресурси, а не фактичного керування ними.

Цей тип управління перехідними процесами ще не можна віднести до управління мережею в повному розумінні цього терміна. Він створюється тільки тоді, коли адміністратор має можливість працювати з єдиним представленням мережі. При цьому відбувся перехід від моніторингу роботи окремих пристроїв до аналізу роботи окремих ділянок мережі, управління її логічною конфігурацією та конкретними робочими параметрами, причому всі ці операції можна було виконувати з консолі управління.

					<b>ВКРБ-123.23.0004.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4



ресурсами, визначити їх призначення, місце в підсистемі та сферу застосування.

2. Виконати детальний аналіз існуючих аналогових систем і аналогових алгоритмів для визначення їх позитивних і негативних якостей для подальшого врахування при розробці аналогових алгоритмів.

3. Розробка функціональної та структурної схем системи, визначення їх складових та обґрунтування вибору елементів підсистеми для управління системою, роботи з ресурсами мережі.

4. Розробити методи та алгоритми забезпечення роботи неоднорідних окремих мереж.

5. Розробити програмне забезпечення для підключення різних окремих мереж.

Об'єктом дослідження є різноспрямовані комп'ютерні мережі.

Тема дослідження – мережеві протоколи та програмні рішення, що забезпечують управління комп'ютерними мережами різного спрямування.

Методи дослідження. Для вирішення поставлених завдань у роботі використовувалися методи теорії інформації, аналізу інформації, методи формування цифрового коду, мережеві протоколи та стандарти.

Практичний сенс. Розроблено програмно-апаратне рішення для забезпечення взаємозв'язку віддалених мереж різного призначення. Система забезпечує можливість об'єднання та централізації управління мережами диспетчеризації, телефонії, відеоспостереження, підприємства та загального призначення, тобто відповідає сучасним вимогам до систем подібного класу.

Таким чином, беручи до уваги вищесказане, система, яка буде розроблена, буде дійсно практичною у щоденній роботі різних функціональних підрозділів. Це дозволить вам замінити досить велику частину щоденної роботи, що виконується вручну, автоматичним виконанням заданих функцій.

Тому тема, яку планується розробити, є дійсно актуальною і система буде затребувана малими та середніми підприємцями.

					<b>ВКРБ-123.23.0004.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

# 1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

## 1.1 Призначення системи

Для ефективної роботи комп'ютерної мережі необхідно централізовано контролювати стан її основних елементів, виявляти та усувати виникаючі проблеми, проводити аналіз продуктивності та планувати розвиток мережі, а також застосовувати сучасні протоколи з урахуванням пластичності при побудові мережі. Визначення інфраструктури з неоднорідними спрямованими зв'язками та інтеграцією віртуального сегмента з фізичним тощо. Цей вид роботи є основним завданням адміністрування мережевої системи.

Адміністрування - управлінські процедури, що керують певними процесами або їх частиною. Як правило, він фіксує та контролює процеси та ситуації, які вимагають цілеспрямованого регулювання або контролю.

Побудова комп'ютерних мереж потребувало управління ними та створеними на їх основі комп'ютерними та інформаційними системами. В результаті з'явилося системне адміністрування.

Основною метою управління системою є адаптація мережі до її намічених цілей і завдань. Ця мета досягається завдяки управлінню мережею, яке мінімізує час і ресурси системного адміністрування, одночасно максимізуючи доступність, продуктивність і продуктивність системи.

У процесі виконання бакалаврської роботи (БР) необхідно проаналізувати сервісні функції та службові програми системного адміністрування роботи з комп'ютерними та мережевими ресурсами, визначити їх призначення, місце в підсистемі та сферу їх застосування.

					<b>ВКРБ-123.23.0004.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

Розроблюване програмне забезпечення призначене для забезпечення виконання наступних сервісних функцій при роботі з ресурсами комп'ютерної мережі: локальної; глобальної, а саме:

- моніторинг доступності хостів;
- сканування «розшарених» ресурсів хоста;
- одержання різнобічної інформації про мережеві пристрої;
- забезпечення безперервності з'єднань;
- використання повного потенціалу підключень.

У програмних засобах системи планується ці функції у вигляді допоміжних програм та модулів, які виконують свою роль в системі. Допоміжні модулі будуть незалежними один від одного і тому зможуть, в разі необхідності, викликатись та працювати одночасно. Їх запуск буде відбуватись автоматично та залежно від критичних подій відбуватиметься перезапуск.

Якщо не вдаватися в подробиці, то завдання, які вирішуються в цій області, поділяються на дві групи: моніторинг роботи мережевих пристроїв і управління роботою мережі в цілому [2]. У першому випадку мова йде про моніторинг окремих мережевих пристроїв (концентраторів, комутаторів, маршрутизаторів, серверів доступу тощо), налаштування та зміни налаштувань, а також видалення нових помилок. Ця досить традиційна робоча група називається Responsive Management. Метою другої групи є моніторинг мережевого трафіку, визначення тенденцій змін і аналіз подій для впровадження системи пріоритезації для забезпечення максимальної продуктивності (проактивне управління). Це також включає зміни конфігурації мережі, керування IP-адресами користувачів, фільтрацію пакетів для забезпечення безпеки даних та багато інших завдань. Ці функції реалізовані в маршрутизаторах, які мають власне обмеження номінального навантаження, але взагалі не мають можливості розширення функцій. Тому ми залишаємо виконання цих завдань ПК, оскільки ці рішення можуть

					<b>ВКРБ-123.23.0004.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>8</b>



Незважаючи на простоту цих концепцій, до них потрібно звикнути. Без нього неможливо зрозуміти суть операційної системи UNIX.

Що стосується користувача, операційна система UNIX спочатку задумувалася як інтерактивна система. Іншими словами, UNIX призначений для термінальної роботи. Для початку роботи необхідно «зайти» в систему, ввівши в безкоштовному терміналі ім'я свого облікового запису (account name) і пароль (password). Особа з іменем облікового запису, зареєстрованим у файлі облікового запису системи, називається зареєстрованим користувачем системи. Реєстрація нових користувачів зазвичай здійснюється системними адміністраторами. Користувачі не можуть змінити ім'я свого облікового запису, але вони можуть встановити та/або змінити свій пароль. Паролі зберігаються в окремому файлі в закодованому вигляді. Якщо користувач забув свій пароль, його не зможуть відновити навіть системні адміністратори!

Інтерфейс користувача – це спосіб традиційної взаємодії з системою UNIX заснований на використанні командних мов (хоча сьогодні графічні інтерфейси користувача стають все більш популярними). Коли користувач авторизується в системі, для нього запускається один із інтерпретаторів команд (відповідно до налаштувань, збережених у файлі /etc/passwd). Загалом, система підтримує кілька програм-оболонки, які мають схожу функціональність, але відрізняються мовами команд. Загальна назва оболонки операційної системи UNIX - оболонка, оскільки кожна оболонка представляє собою зовнішнє середовище ядра системи [5].

Як і у випадку з привілейованим користувачем, ядро UNIX ідентифікує кожного користувача за його ідентифікатором, який є глобально унікальним значенням, яке надається користувачеві під час входу в систему. Крім того, кожен користувач належить до групи користувачів, яка також ідентифікується цілим числом (GID - Group Identifier). Зрозуміло, що системний адміністратор, який також є зареєстрованим користувачем,

					<b>ВКРБ-123.23.0004.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>10</b>



Програми - OS UNIX є операційним середовищем для використання існуючих програм і середовищем для розробки нових програм. Нові програми можуть бути написані на різних мовах (Fortran, Pascal, Modula, Ada та ін.). Однак стандартною мовою програмування в середовищі операційної системи UNIX є C (який останнім часом все частіше замінюється на C++). Це пояснюється тим, що, по-перше, сама система UNIX написана на мові Cі, а по-друге, мова Cі є однією з найкращіших стандартизованих мов.

Команди. Кожна мова команд сімейства оболонок насправді складається з трьох частин: службових структур, які дозволяють вам маніпулювати текстовими рядками та створювати складні команди з простих команд; вбудовані команди, що виконуються безпосередньо інтерпретатором командної мови; Команди представлені окремими виконуваними файлами.

Процеси в операційній системі UNIX — це програми, які виконуються у власному віртуальному адресному просторі. Коли користувач входить в систему, автоматично створюється процес для запуску програми оболонки. Коли інтерпретатор команд зустрічає команду, яка відповідає виконуваному файлу, він створює новий процес і запускає в ньому відповідну програму, починаючи з основної функції. Ця запущена програма може, у свою чергу, створити процес і запустити в ньому іншу програму (вона також повинна містити основну функцію).

Механізм перенаправлення вводу/виводу є одним із найелегантніших, потужних і водночас найпростіших механізмів операційної системи UNIX. Мета цього механізму полягає в наступному: оскільки UNIX є інтерактивною системою, звичайні програми вводять текстові рядки з терміналу та виводять отримані текстові рядки на екран терміналу. Щоб забезпечити більш гнучке використання таких програм, бажано надавати їм вхідні дані з файлу або з вихідних потоків інших програм і направляти їхній вихід у файл або у вхідні потоки інших програм [ 3 ] .

Таким чином, підсистема, що підлягає розробці, буде багатофункціональною та дійсно корисною. Тому область її впровадження надзвичайно велика. Її можна охарактеризувати одним словом – всюди. Всюди, де є необхідність роботи з даними на базі комп'ютерних технологій з використанням мереж будь-якого типу.

Кафедра \_ КБПЗ \_ 2023 рік

					ВКРБ-123.23.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13



знання мережевих технологій і протоколів, багато організацій, навіть працюючи з мережевим обладнанням інших компаній, визнають цінність професійних сертифікатів CISCO. Зокрема, сертифікація CCIE є однією з найбільш визнаних і визнаних в комп'ютерній індустрії. Але, окрім ціни на апаратне забезпечення, нам необхідно отримати сертифікацію програмного забезпечення через багаторівневість організації.

Другий підхід ширший, але на перший погляд менш досконалий. Це маршрутизація SoftLinux. Спеціальне обладнання чи сертифікація програмного забезпечення не потрібні. Неочевидно, що його непромислова версія використовується в звичайних пристроях, таких як маршрутизатори таких компаній, як TP-Link, D-Link, Asus, Xiaomi, GrandStream тощо. Під власною оболонкою вони використовують рішення SoftLinux, які завдяки ліцензії GNU дозволяють значно здешевити продукцію компанії, що жодним чином не впливає на якість, а навпаки, дає можливість для значущого вимірювання функціональності продуктів. пристроїв, додайте Наприклад, додайте порти USB для знімних носіїв, модемів або принтерів. Цей прийом використовується, наприклад, у стандартних смартфонах (режим модему або точки доступу).

Тому, виходячи з вищевикладеного, розглянемо розробку системи маршрутизації на основі рішень SoftLinux.

Переваги:

- Нижча ціна;
- вибір параметрів масштабування;
- немає необхідності платити за ліцензію;
- Різні технічні рішення.

На ринку програмних продуктів в наявності чимало програмної продукції аналогічного спрямування. Тому доцільно розглянути наявні представники, щоб визначитись з концепцією розробки програмного забезпечення, структурою його побудови та визначити позитивні і негативні якості з ціллю подальшого врахування при розробці ПЗ.

На початку пошуку та огляду програм аналогічного спрямування було виявлено, що фактично багатofункціональні автоматизовані системи, які б

					<b>ВКРБ-123.23.0004.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

забезпечували виконання багатьох функцій в одній програмі та їх одночасну роботу з різними мережними ресурсами на ринку програмних продуктів відсутні.

Тому аналіз було виконано в розрізі окремих програмно-апаратних пристроїв та програм.

Nmap – різнобічний сканер портів і засіб ідентифікації операційної системи (ОС). Це без будь-яких сумнівів один з кращих сканерів портів. Його головний автор – програміст з псевдонімом “Fyodor”, розробки якого використовуються в багатьох інших програмах та портирувані практично на всі ОС, що використовуються. На Nmap спирається сканер Nessus. Доступні також декілька додатків, включаючи програму Nlog. Nmap на даний час є складовою інструментарію кожного адміністратора безпеки [7]. Програма Nmap має наступні позитивні якості:

- у неї багато опцій. Прості сканери портів доступні з такими засобами, як Sam Spadl, але в порівнянні з ними у Nmap – величезна кількість опцій, що надає можливість необмеженого числа варіантів сканування мережі. Можливо знизити частоту відправлення зондувальних пакетів чи, навпаки, збільшити її, якщо є запас ширини смуги пропускання. Опції невидимості – ще один елемент репертуару Nmap. За їх допомогою перевіряються наскільки чутлива система до виявлення вторгнень. Nmap, виходячи за рамки простого сканування портів, дозволяє виконувати ідентифікацію ОС, що є дуже позитивним фактором при встановленні відповідності між IP-адресами та машинами;

- вона легка та потужна. Код Nmap невеликий і буде виконуватись навіть на самих застарілих машинах. В невеличкому обсязі програмного коду схована величезна потужність та добра оптимізація алгоритму, що без проблем сканує дуже великі мережі;

- вона проста в використанні. Існує багато способів запуску програм Nmap. Програма добре документована, для Unix та Windows має графічний інтерфейс, легко сприймається навіть недосвідченим користувачем.

До недоліків Nmap слід віднести наступні:

					<b>ВКРБ-123.23.0004.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

- орієнтація на виконання тільки однієї функції – сканування портів та мережі;
- надто великий перелік сервісних дій, які залишаються поза увагою користувача;
- дуже велика ціна продукції із-за великого сервісу дій, який в багатьох випадках не потрібний в використанні.

Samba – програмне забезпечення (в подальшому – програма), що призначене для адміністраторів мереж [5]. Програма дозволяє перегляд розшарених ресурсів в локальній мережі, сканування діапазонів IP-адрес на наявність сервісів загального використання.

По результатах роботи Samba надає докладні звіти користувачу по NetBios. Програма також дозволяє переглянути список комп'ютерів, що знаходяться в мережі, і користувачів, які зареєстровані на комп'ютерах.

Недоліки, присутні Nmap, відносяться і до Samba.

Пристрій NetPing призначений для перевірки працездатності сегментів мереж Ethernet, в яких відсутні пристрої зі статичною (постійною) IP-адресою [3]. Зазвичай в таких мережах не існує можливості гарантовано діагностувати працездатність сегментів, але введення до складу компонентів мережі NetPing дозволяє суттєво поліпшити ситуацію.

Пристрій NetPing є мережним обладнанням з постійною IP-адресою, яке весь час функціонує в даному сегменті мережі, незалежно від того, чи діють в даний момент часу інші мережні пристрої (комп'ютери, мережні робочі станції, мережні принтери, тощо). Це дозволяє віддалено по необхідності протестувати працездатність конкретної ділянки мережі або ж використати системи постійного моніторингу (за рахунок послання ICMP пакетів до пристрою і моніторингу відповідей на них).

Пристрій NetPing комплектується програмою NPCFG за допомогою якої значно розширює діапазон виконуючих функцій: віддалено змінює свої параметри з любого комп'ютера в мережі, забезпечує самозахист від зміни параметрів.

					<b>ВКРБ-123.23.0004.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17



Jetico Personal Firewall організовує три рівні захисту: перевірку та фільтрацію мережевих пакетів, дій мережевих програм і процесів користувача [15].

Sunbelt Kerio Personal Firewall — безкоштовна програма захисту від хакерського проникнення та витоку даних.

Недоліком є також і апаратно-програмне виконання. Підключення приладу в мережу потребує обслуговування та додаткових енерговитрат, наявності відповідних до паспорту кліматичних умов експлуатації.

Таким чином, в результаті проведеного в період проходження переддипломної практики аналізу систем-аналогів та проведеного в даному розділі їх розгляду, були виявлені наступні недоліки:

- орієнтація на виконання обмеженої кількості функцій в одному напрямку;
- ПЗ надає великий перелік сервісних дій, які залишаються поза увагою користувача, яких необхідно уникнути при розробці ПЗ нашої підсистеми.

Організація процесу збору, обробки та передачі інформації повинна відповідати та здійснюватися відповідно до чинних державних стандартів, які встановлюють низку загальних та спеціальних вимог, а саме:

- необхідно використовувати сучасні методи організації та управління, включаючи системний структурний аналіз і моделювання процесів, економіко-математичні методи;
- слід використовувати великі параметричні діапазони та раціональні стандарти виробів комп'ютерної техніки;
- має бути спрямований на рівень автоматизації збору, підготовки, обробки, передачі та подання оптимальної інформації для конкретних обставин.

					<b>ВКРБ-123.23.0004.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>19</b>





Принцип єдності даних дозволяє комбінувати групи даних, операційні системи та мови введення та виведення для опису компонентів системи.

Принцип незмінності означає, що система, яка розробляється, повинна бути універсальною або типовою з точки зору використання для інших вимог. Вирішення цієї проблеми планується шляхом включення в програмне забезпечення загальних або спеціальних модулів. Достатня кількість модулів даного типу в структурі програмного забезпечення забезпечує повноту усунення неполадок системи.

## **2.2 Обґрунтування вибору засобів для побудови системи та мови програмування**

Для розробки підсистеми маршрутизації в якості платформи обираємо функції ОС Linux та можливості фаєрволу IPTables, що представляє собою між мережний екранний фаєрвол з широким функціоналом [1].

Коли пакет надходить до нашого мережевого монітора, він спочатку потрапляє на мережевий пристрій, перехоплюється відповідним контролером, а потім пересилається до ядра. Потім пакет проходить через кілька таблиць, а потім надсилається локальною програмою або пересилається іншому одержувачу.

Iptables — це інструмент командного рядка, який був стандартним інтерфейсом керування брандмауером для ядер Linux, починаючи з версії 2.4. Для служби iptables потрібні права root. Попередниками iptables були проекти ipchains (застосовувався до керування брандмауером ядра Linux версії 2.2) та ipfwadm (для ядра Linux версії 2.0). Iptables підтримує ідеологію, похідну від ipfwadm: поведінка брандмауера визначається правилами, кожне з яких складається з критерію та дії, що виконується з пакетами, які відповідають цьому критерію. Концепція ланцюжків - незалежних списків правил - з'явилася в ipchains. Було введено окремі ланцюжки для фільтрації пакетів INPUT, OUTPUT і FORWARD. Як

					<b>ВКРБ-123.23.0004.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

продовження цієї ідеї в iptables з'явилися таблиці - групи незалежних ланцюжків. Кожна таблиця виконувала власне завдання — ланцюжки таблиць фільтрів відповідали за фільтрацію, ланцюжки таблиць NAT — за перетворення мережевих адрес, завдання таблиці mangle включали інші зміни заголовків пакетів (наприклад, TTL або TOS). Також була дещо змінена логіка роботи ланцюга: у біпчейнах усі вхідні пакети, включаючи транзитні, проходили через ланцюг INPUT. в iptables лише пакети, адресовані самому хосту, проходять через INPUT. Ця функція дозволила iptables використовувати інформацію про все з'єднання під час обробки окремих пакетів. Тут iptables набагато кращий, ніж ipchains. iptables може відстежувати стан з'єднання та перевіряти, змінювати або фільтрувати пакети не лише за їхніми заголовками (джерело, призначення) або вмістом пакетів, а й за контактною інформацією. Можна сказати, що iptables аналізує не тільки дані, що передаються, але й контекст їх передачі, і тому може приймати обґрунтовані рішення щодо долі кожного окремого пакета.

Ця функція, поряд з іншими сервісами, дозволяє створювати розгалужені шляхи і з'єднувати різні віддалені мережі, створювати необхідну топологію мережі з урахуванням стандартів, критеріїв безпеки, підвищувати надійність і автоматизувати деякі процеси.

### 2.3 Розгорнута постановка завдання

В процесі розробки БР необхідно виконати наступний обсяг робіт:

- а) визначити та обґрунтувати актуальність теми БР;
- б) визначити призначення системи та область її впровадження;
- в) аналіз існуючих аналогових алгоритмів, аналогових систем з метою виявлення їх позитивних і негативних властивостей. Результати аналізу будуть враховані при подальшій розробці;

					<b>ВКРБ-123.23.0004.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>23</b>

г) розробка та обґрунтування методу побудови системи, визначення принципів розробки. Розробка функціональної та структурної схем системи. Визначте його складові та обґрунтуйте свій вибір;

д) розробка системного програмного забезпечення, яке дає змогу реалізувати завдання КТ. Створювати основні блок-схеми програми та алгоритми підпрограм;

е) розробляє заходи щодо введення підсистеми в промислову експлуатацію;

є) надати стислу характеристику розробленої підсистеми за всіма основними параметрами;

ж) надати перелік скорочень, символів і спеціальних термінів, що будуть використані при розробці БР;

з) навести список літератури, що використалась при розробці БР;

і) розробити анотацію на українській та англійській мовах.

					ВКРБ-123.23.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

## 3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

### 3.1 Опис функціонування системи

Темою БР передбачена розробка програмного забезпечення для системного адміністрування неоднорідної мережі на базі Soft Linux рішень, яка забезпечує взаємодію віддалених мереж з різним призначенням.

Після проведення аналізу існуючих систем аналогів, програм-аналогів та виявлення їх позитивних сторін було вирішено розробити ПЗ з урахуванням виявлених позитивних якостей та за виключенням недоліків для рішення поставленої технічним завданням задачі.

В процесі оцінки задач, які повинні забезпечувати поставлену мету по розробці системи виділяємо основні критерії її функціонування

- одночасне використання декількох каналів Інтернету;
- об'єднання віддалених мереж;
- сканування ресурсів мережі з вільним доступом;
- виведення різнобічної інформації про мережні пристрої;
- аутентифікація користувача;
- обробка системних помилок;
- виведення інформації про діяльність системи, ПК.

Наступна з вище означених функцій, а саме: одночасне використання декількох каналів Інтернету не є стандартним функціоналом будь яких маршрутизаторів, але дана функція забезпечить безперебійність зв'язку різних частин нашої системи. Реалізація даного рішення варіативна, тобто можемо використовувати обидва канали з автоматичним балансуванням, або виділяти один з каналів для технічних мереж (приладів автоматизації, відеонагляду, телефонії тощо.)

Проектні рішення, які будуть використовуватися, повинні виконувати всі перераховані вище функції. Крім того, вони повинні бути компактними і зрозумілими і відповідати сучасним вимогам до систем подібного класу. Це

					<b>ВКРБ-123.23.0004.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25





SNAT (Source Network Address Translation) використовується для зміни вихідних адрес пакетів. За допомогою цієї дії ви можете приховати структуру локальної мережі і в той же час поділити зовнішню IP-адресу між комп'ютерами локальної мережі для доступу в Інтернет. У цьому випадку брандмауер автоматично виконує пряме і зворотне перетворення адрес за допомогою SNAT, що дозволяє комп'ютерам локальної мережі підключатися до серверів Інтернету.

Маска (MASQUERADE) використовується з тією ж метою, що і SNAT, але на відміну від останнього, MASQUERADE дає більше навантаження на систему. Тому що, коли запитується ця дія, запит IP-адреси виконується для мережевого інтерфейсу, зазначеного в дії, тоді як у випадку SNAT IP-адреса вказується безпосередньо. Однак через цю різницю MASQUERADE може працювати у випадках динамічної IP-адреси.

Таблиця фільтрів — це таблиця, яка повинна містити набір правил для виконання фільтрації пакетів. Залежно від вмісту пакети можуть бути перенаправлені або відхилені (дії ACCEPT або DROP). Звичайно, ми можемо фільтрувати пакети і в інших таблицях, але ця таблиця існує саме для цілей фільтрації. Більшість існуючих дій можна застосувати до цієї таблиці, але деякі з дій, розглянутих раніше в цьому розділі, слід виконувати лише для власних таблиць.

Як ми вже дізналися, пакети в просторі ядра можуть мати кілька різних станів залежно від типу протоколу. Однак пакети можуть мати лише 4 стани за межами ядра. В основному стан пакета використовується за критерієм - стан. Можливі стани: новий, встановлений, підключений і недійсний. Кожен можливий стан описано в таблиці 3.1 [8]. Ці чотири умови можна використовувати в критерії умови. Механізм визначення стану дозволяє створити надзвичайно міцний і ефективний захист.

					<b>ВКРБ-123.23.0004.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>28</b>

Таблиця 3.1. Ознаки різних станів у користувача

Стан	Опис
NEW	<p>Прапор NEW вказує, що пакет є першим для цього з'єднання. Тобто це перший пакет, який бачить модуль стеження. Наприклад, якщо отримано SYN і цей пакет є першим пакетом для цього з'єднання, він отримає статус НОВИЙ. Однак пакет може не бути SYN-пакетом і мати статус NEW. У деяких випадках це може спричинити певні проблеми, але також може бути дуже корисним, наприклад, під час спроби «відновити» з'єднання, «втрачені» іншими брандмауерами, або у випадках, коли час очікування з'єднання минув, але саме з'єднання не закрито</p>
RELATED	<p>ПОВ'ЯЗАНА ситуація є однією з тих «важких». З'єднання отримує статус RELATED, якщо воно пов'язане з іншими з'єднаннями з атрибутом ESTABLISHED, тобто коли воно запускається з уже встановленого з'єднання з атрибутом ESTABLISHED. Хорошим прикладом з'єднання, яке можна вважати АСОЦІЙОВАНИМ, є з'єднання даних FTP, підключене до контрольного порту FTP, а також з'єднання DCC, що походить від IRC. Слід зазначити, що більшість протоколів TCP і деякі протоколи UDP досить складні й передають інформацію про з'єднання через поле даних пакетів TCP або UDP, тому для належної роботи потрібні спеціальні допоміжні модулі.</p>
ESTABLISHED	<p>Статус ESTABLISHED вказує на те, що це не перший пакет у з'єднанні. Процедура встановлення стану СТАНОВЛЕНО досить проста для розуміння. Єдина вимога для з'єднання – перейти в стан ESTABLISHED, вузол мережі повинен надіслати пакет і отримати відповідь від іншого вузла (хосту). Після отримання відповіді статус НОВИЙ або КОНТАКТ стане ВСТАНОВЛЕНИМ.</p>
INVALID	<p>Прапор INVALID вказує на те, що пакет не може бути розпізнаний і тому не може мати спеціальний статус. Це може статися з кількох причин, наприклад, брак пам'яті або отримання повідомлення про помилку ICMP, яке не відповідає жодному відомому з'єднанню. Можливо, найкращим варіантом є використання дії DROP для таких пакунків.</p>

Раніше всі порти вище 1024 потрібно було відкрити, щоб повернутися до локальної мережі, але тепер із сучасним механізмом це більше не потрібно, оскільки доступ можна лише «відкрити». повторно відправити (відповісти) трафік, зупинити спроби зовнішнього підключення.

TCP-з'єднання завжди встановлюється шляхом надсилання трьох пакетів, які ініціюють і встановлюють з'єднання, через яке дані будуть передаватися в майбутньому. Сеанс починається з передачі SYN-пакета, у відповідь на який надсилається SYN/ACK-пакет, а ACK-пакет підтверджує встановлення з'єднання. Після цього з'єднання вважається встановленим і готовим до передачі даних. Може виникнути питання: «Як стежити за спілкуванням?». Насправді все досить просто.

Відстеження майже однакове для всіх типів з'єднань. На малюнку 3.1 показано всі етапи встановлення з'єднання. Як бачите, трекер насправді не відстежує хід підключення з точки зору користувача. Все просто, як тільки глядач «побачить» перший пакет (SYN), він присвоїть йому статус **НОВИЙ**. Коли другий пакет (SYN/ACK) проходить через трекер, з'єднання отримує статус **ВСТАНОВЛЕНО**. Чому другий пакет? Налаштувавши набір правил, ви можете дозволити пакетам зі статусом **НОВИЙ** і **ВСТАНОВЛЕНИЙ** залишати локальну мережу, лише пакетам зі статусом **ВСТАНОВЛЕНО** пропускати вхідний трафік, і все буде добре. І навпаки, якщо трекер продовжує обробляти з'єднання як **НОВЕ**, тоді він більше не зможе спілкуватися із "зовнішнім світом" або повинен буде дозволити **НОВІ** пакети через локальну мережу. З точки зору ядра все здається складнішим, оскільки в просторі ядра TCP з'єднання мають ряд проміжних станів, яких немає в просторі користувача. Загалом вони відповідають специфікації RFC 793 - Transmission Control Protocol [8].

					<b>ВКРБ-123.23.0004.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>30</b>



Таким чином, маємо всі необхідні дані для побудови структурної і функціональної схем підсистеми.

### 3.2 Розробка структурної схеми

Розроблюваний програмний продукт – підсистема для управління та адміністрування неоднорідної мережі на базі Soft Linux рішень повинна відповідати заданим технічним вимогам і забезпечувати такі терміни виконання операцій, щоб вони не виходили за рамки технологічного процесу, який виконується.

Розроблена підсистема побудована по модульному принципу, тобто є головний, зв'язуючи модуль та підлеглі модулі, кожен з яких виконує якусь одну функцію.

Головним модулем виступає мережний екранний фаєрвол IPTables при розробці правил внесені змінні, для спрощення записів, та записи які формують маршрути на базі ядра, що дає можливість уникнути додаткових скриптів.

Таким чином, розробник може без особливих труднощів допрацювати записи, виправити можливі помилки, додати або вилучити якісь нові вузли мережі не перезавантажуючи систему.

Система, що підлягає розробці, буде виконувати багато різних операцій при роботі з мережею. Тому потрібно чітко розуміти топологію мережі яку будуємо. Припустимо, що ми маємо об'єднану систему та поділимо її на частини:

- головний канал Інтернету;
- резервний канал Інтернету;
- сервер маршрутизації;
- сервери сервісів (корпоративний сегмент);
- сервери обліку (промисловий сегмент);
- демілітаризована мережа WiFi;
- персональні комп'ютери;
- прилади (промисловий сегмент).

Сформуємо та розглянемо структурну схему підсистеми, що підлягає розробці в процесі проектування, яка показана на рисунку 3.5.

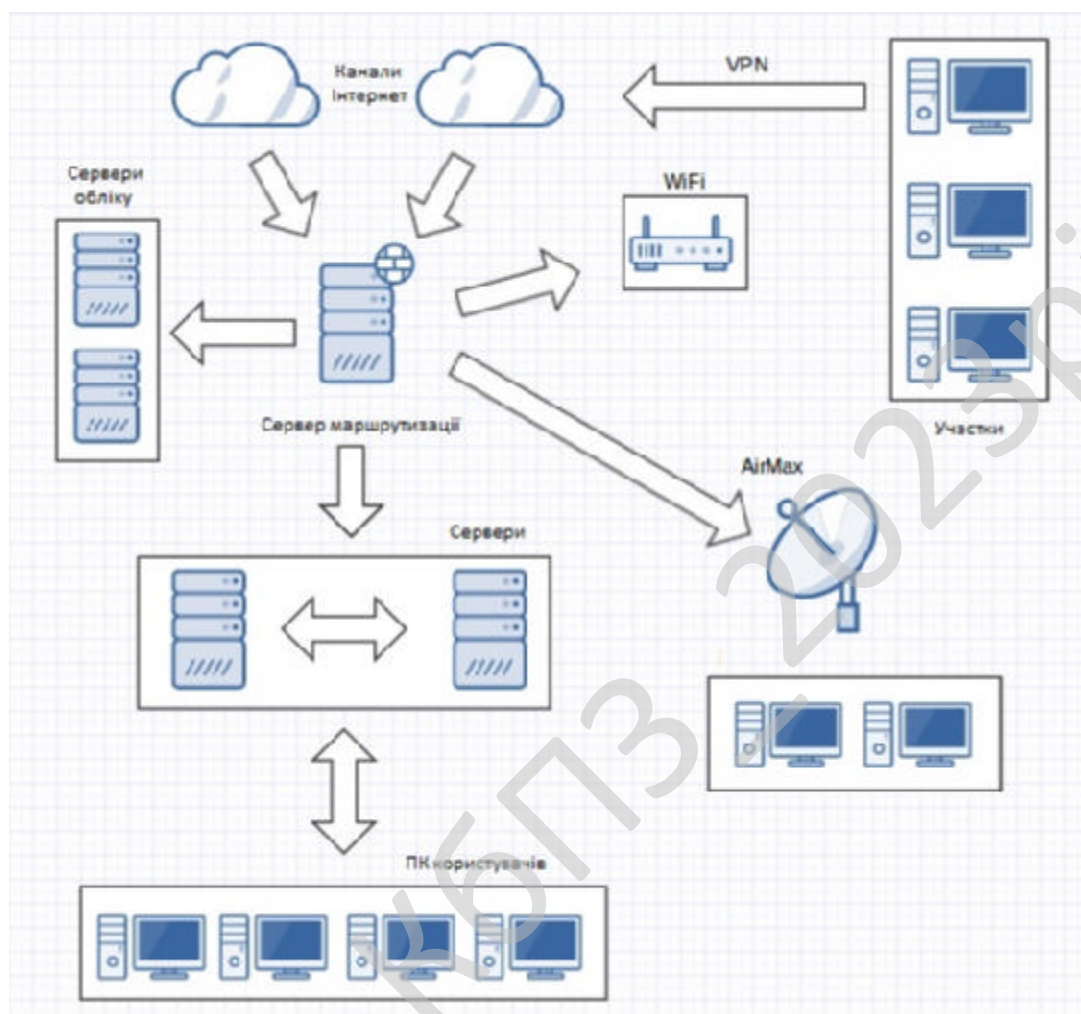


Рисунок 3.2 – Структурна схема системи

Технологічно обумовлено наявність зовнішніх Інтернет адрес на віддалених ділянках, тому виключаємо використання віртуальних персональних мереж тобто VPN, кожен об'єкт буде мати правила, які маршрутизують мережі по зовнішнім адресам та визначають достовірність завдяки сертифікатам. Сегмент системи який використовує обладнання AirMax будимо вважати частиною системи на кшталт прямого з'єднання по кабелю, оскільки дане обладнання реалізує мостове з'єднання.

### 3.3. Розробка функціональної схеми

На відміну від структурної, функціональна схема більш детально відображає зв'язки між ланками мережі та визначає її обмеження і потреби.

Потреби:

- трансляція портів зовнішньої Інтернет адреси до поштового серверу;
- трансляція портів зовнішньої Інтернет адреси до серверу обліку;
- доступ до мережі Інтернет для серверу авторизації;
- доступ до мережі Інтернет для серверу терміналів;
- доступ до мережі серверу баз даних для серверу терміналів;
- доступ до мережі Інтернет для ПК користувачів (за потреби);

Обмеження:

- ізоляція від корпоративної мережі серверів резервних копій;
- ізоляція корпоративної від гостьової мережі.

Реалізація маршрутизації на віддалених дільницях буде забезпечуватися ПК оператора, на якому встановлена операційна система Linux з графічною оболонкою. Таке рішення забезпечить економію на додаткових пристроях та ліцензуванні.

Система, що підлягає розробці, буде забезпечувати виконання всіх функцій, що закладені в підсистемі. Мережевий екран забезпечить також одночасну роботу з різними мережними ресурсами. Саме це проектне рішення визначає високу ступінь новизни розробки.

Маємо всі необхідні дані для побудови функціональної схеми підсистеми, яка представлена на рисунку 3.3.

					<b>ВКРБ-123.23.0004.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

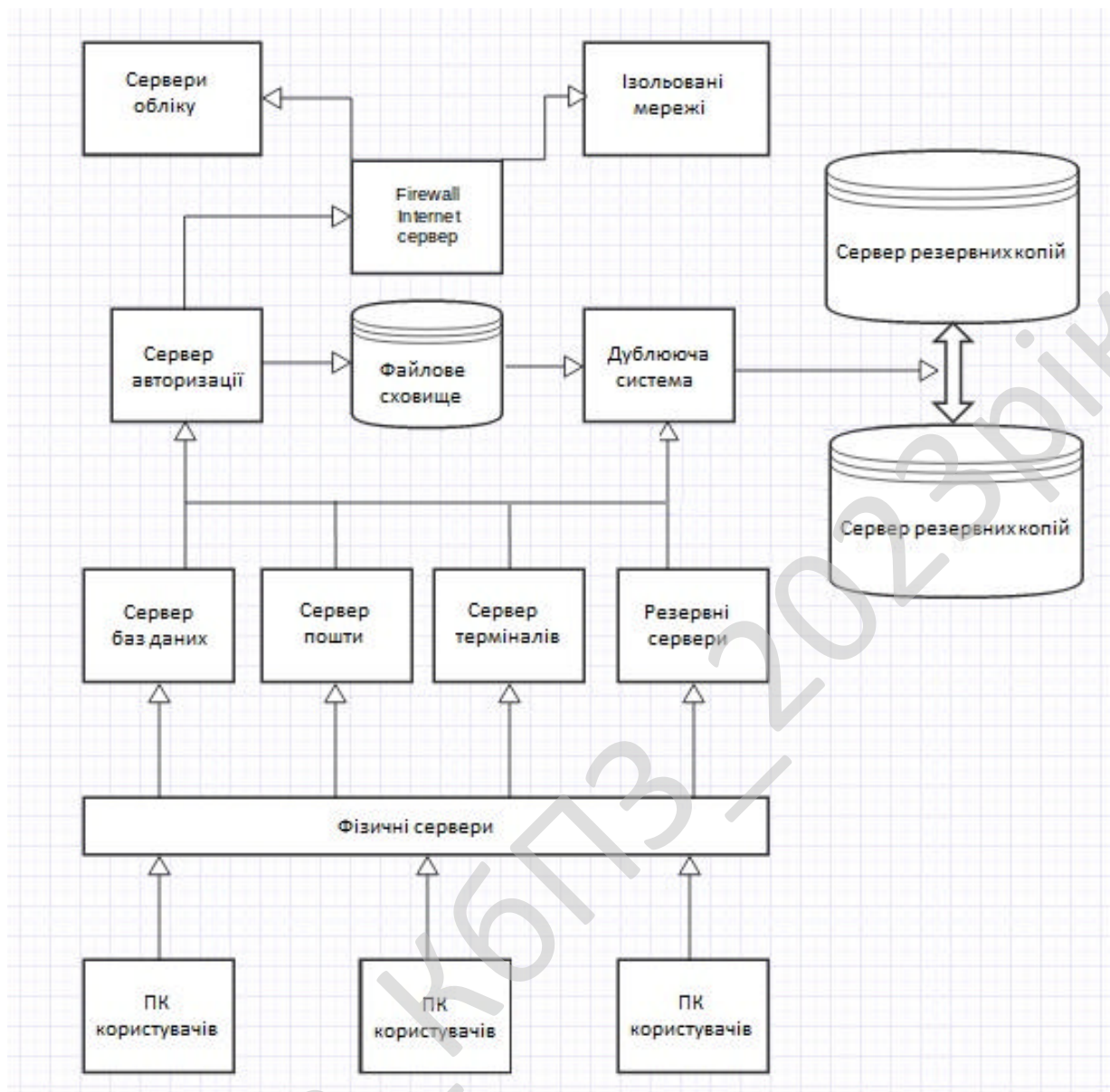


Рисунок 3.3 – Функціональна схема роботи системи

### 3.4. Розробка діаграми процесів.

Для більш детального та наочного пояснення принципів роботи системи розроблена та наводиться на рисунку 3.4 діаграма процесів.

При старті операційної системи виконується запуск файлу який містить головні директиви об'єднаної мережі з урахуванням функціональних умов сегментів мережі. Частково використовуючи систему віртуалізації вводимо не керований комутатор для об'єднання в мережу серверу баз даних та термінального серверу. Резервуючий сервер комутується з головним



Таким чином, визначивши та обґрунтувавши:

- проектні рішення щодо побудови системи та конфігурування її програмного забезпечення;
- побудову структурної та функціональної схем системи;
- теоретичну побудову основних принципів маршрутизації;
- обмеження, вимоги, умови та критерії, яких повинна дотримуватись система при виконанні встановлених функцій;
- комплекс технічних засобів, що забезпечить експлуатацію системи;
- перелік вхідних даних, необхідних для організації і контролю, роботи системи.

Тобто маємо всі необхідні критерії та дані для розробки підсистеми для управління та адміністрування неоднорідної мережі.

					<b>ВКРБ-123.23.0004.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

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

### 4.1 Розробка блок-схем та опис алгоритмів функціонування системи

При розробці програмного забезпечення була здійснена спроба уникнути негативного досвіду, який був виявлений в процесі аналізу систем-аналогів, та взяти за основу в побудові програмного забезпечення підсистеми виявлені позитивні якості.

В розробці підсистеми використане програмування під ОС Linux та пакети прикладних програм та модулів. Оскільки різниця між представником серверної гілки та користувацької дуже незначна і деякі гілки глобального напрямлення взагалі мають комбіновану структуру, ми будемо використовувати linux ubuntu server без графічної оболонки з компонентами її підтримки, завдяки чому ми матимемо можливість запускати модулі та програми віддалено, що називається “прокиданням” програми, використовуючи ssh-з’єднання та тунелювання. Такий підхід надасть можливість контролювати систему навіть зі смартфона, використовуючи шифрований ключ доступу.

Розроблене в процесі виконання БР програмне забезпечення підсистеми забезпечує виконання сервісних функцій:

- одночасне використання декількох каналів Інтернету;
- об’єднання віддалених мереж;
- сканування ресурсів мережі з вільним доступом;
- виведення різнобічної інформації про мережні пристрої;
- аутентифікація користувача;
- обробка системних помилок;
- виведення інформації про діяльність системи, ПК.

					ВКРБ-123.23.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

Система спроектована у відповідності з вимогами та концепцією побудови систем аналогічного напрямлення та класу, що визначені та описані в розділі 2 пояснювальної записки. По суті система є багатофункціональною підсистемою реального часу, оскільки обробка даних здійснюється за позначками реального часу.

При підготовці програмного забезпечення підсистеми використаємо наступні принципи побудови систем аналогічного спрямування:

- основні принципи організації збирання і обробки інформації;

Реалізуємо це положення у вигляді єдиної інтегрованої мережі збору, накопичення і обробки інформації. Користувач буде мати доступ до вихідної інформації завдяки вбудованій системі логуювання. Необхідно також застосувати глобальну мережу передачі даних для забезпечення зв'язку між обробкою даних на верхніх та на нижніх рівнях системи. При побудові мережі використовуємо стандартні методи та стандартні компоненти; основні принципи організації контролю. При розробці ПЗ уводимо процедури перевірки достовірності інформації та якості виконання запланованих функцій на всіх рівнях і в усіх точках підсистеми, де здійснюється облік і обробка даних. Це важливо не тільки з технічної точки зору (мається на увазі сто відсоткове забезпечення реалізації функціональних вимог до підсистеми), але також і з точки зору економічних і правових взаємовідносин виробника, постачальника і споживача.

Окрім цього, розроблена в процесі виконання БР підсистема виконує задачу контролю в автоматичному режимі і забезпечує виконання та перевірку наступних параметрів:

- виконання заданих функцій;
- забезпечення одночасної роботи вище означених функцій з різними мережними ресурсами;
- запуск кожної окремої функції, що являються предметом нашої розробки, в довільному порядку по факту виникнення потреби.

					<b>ВКРБ-123.23.0004.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>39</b>

Таке представлення підсистеми є зручним для виконання визначених згідно ТЗ функцій роботи системи. Але для побудови дійсно діючого спроможного ПЗ необхідно більш детально розглянути теоретичні аспекти цього питання та виконати необхідні розрахунки реалізації проекту.

Як було визначено у розділі 3 пояснювальної записки, розроблена система має в своїй структурі наступні складові: систему маршрутизації та аунтифікації віддалених підключень, представлені фаєрволом, налаштування використання декількох мереж Інтернету, систему віртуальної маршрутизації в системі віртуалізації.

Програмне забезпечення системи має складну структуру побудови: головна програма та ряд програм-модулів, що забезпечують виконання функцій. Розглянемо блок-схему роботи алгоритму головної програми на рисунку 4.1

Алгоритм роботи системи маршрутизації з одночасним використанням декількох каналів Інтернет:

Крок 1: Скидання таблиці маршрутизації;

Крок 2: Оголошення умов вхідних правил на мережеві інтерфейси та порти;

Крок 3: Оголошення віддаленого доступу та захист від несанкціонованого втручання;

Крок 4: Оголошення первинних заборон та обмежень;

Крок 5: Оголошення прокидання вхідних портів;

Крок 6: Оголошення наявних мереж;

Крок 7: Оголошення дозволів та обмеження користувачів різного рангу;

Крок 8: Маскування користувачів головного та резервного каналу Інтернету;

					<b>ВКРБ-123.23.0004.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>40</b>

Крок 9: Визначення стану інших маршрутів залежно від протоколу;

Крок 10: Визначення стану інших з'єднань.

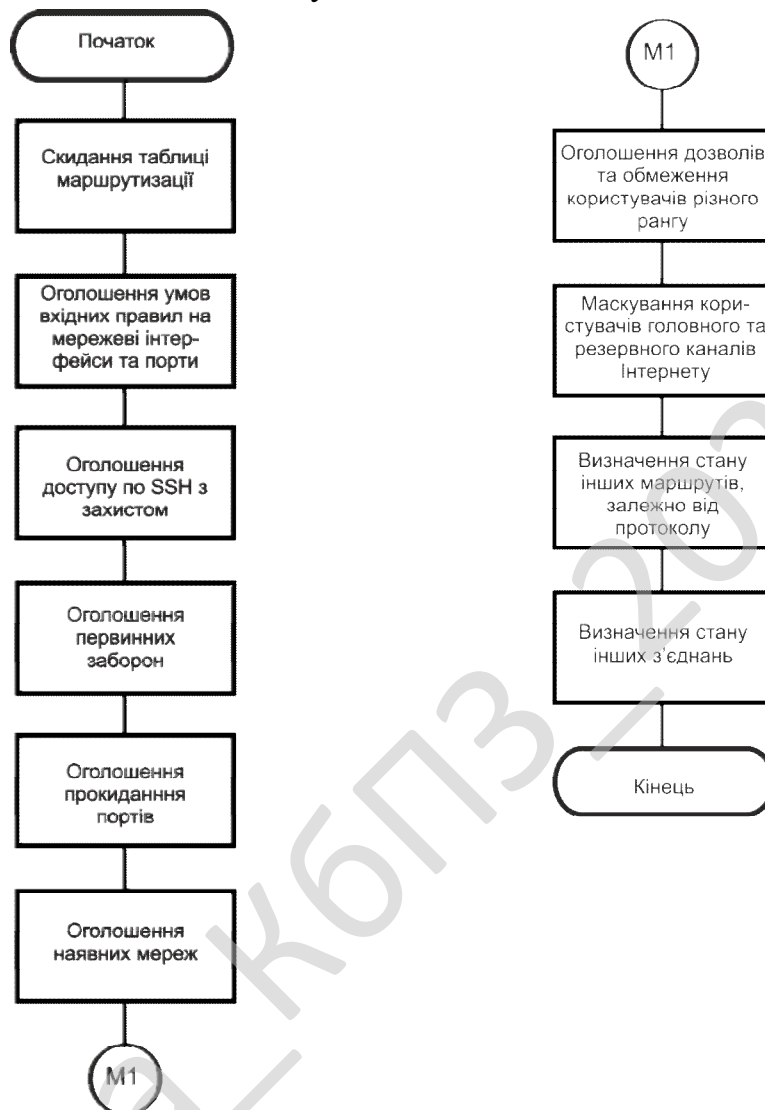


Рисунок 4.1 - Блок-схема головного алгоритму роботи системи маршрутизації з одночасним використанням декількох каналів Інтернет

Після визначення стану інших з'єднань, відбувається розподіл пакетів по каналам Інтернет, згідно алгоритму, наведеного на рисунку 4.2.

Алгоритм працює наступним чином:

Крок 1: Передача пакету;

Крок 2: Перевірка, чи задовольняє пакет правилам вхідної маршрутизації. Якщо не задовольняє, повертаємося до кроку 1. Якщо задовольняє, переходимо до наступного кроку;

Крок 3: Перевірка чи задовольняє пакет правилам каналу. Якщо задовольняє, передаємо пакет по першому каналу Інтернет. Якщо не задовольняє, передаємо пакет по другому каналу.

Таким чином задіяні обидва канали Інтернет (головний та резервний), які працюють у комплексі як один. Що дає підвищення пропускнуої спроможності каналу передачі даних мережі.



Рисунок 4.2 - Блок-схема алгоритму розподілення пакетів між каналами Інтернету

Розроблений алгоритм та його опис – програмний код досить, компактний завдяки використанню циклічної обробки події і тому легкий та зрозумілий для сприйняття.

Таким чином, були визначені та обґрунтовані:

- проектні рішення щодо побудови підсистеми та розробки її програмного забезпечення;
- побудова структурної та функціональної схем системи;
- теоретична побудова основних алгоритмів;
- обмеження, вимоги, умови та критерії, яких повинна дотримуватись система при виконанні встановлених функцій;
- комплекс технічних засобів, що забезпечить експлуатацію системи;
- перелік вхідних даних, необхідних для організації, контролю та роботи системи.

Таким чином, маємо всі необхідні критерії та дані для розробки програмного забезпечення системи.

Лістинг розробленого програмного забезпечення, виконаного на базі Soft Linuks рішень, для реалізації об'єднання віддалених різнорідних мереж і для централізації управління систем диспетчеризації та інших об'єктів загального призначення, представлений у додатку Б.

У першу чергу розглянемо реалізацію правил маршрутизації одночасно по двом каналам Інтернет:

IF1=eth0

IF2=eth1

IF — Мережеві інтерфейси двох провайдерів.

IP1=XXX.XXX.XXX.XXX

IP2=YYY.YYY.YYY.YYY

					<b>ВКРБ-123.23.0004.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

IP — зовнішні IP-адреси, котрі видають провайдери.

P1=XXX.XXX.XXX.XXX

P2=YYY.YYY.YYY.YYY

P — шлюзи від провайдерів.

Policy routing дозволяють використовувати дві різні таблиці маршрутизації в залежності від адрес які видають провайдери:

SRV11=192.168.0.254

SRV12=192.168.0.253

SRV11 та SRV12 дві внутрішні адреси серверу, які використовуються для маршрутизації від обох провайдерів:

```
#echo «101 T1» >> /etc/iproute2/rt_tables
```

```
#echo «102 T2» >> /etc/iproute2/rt_tables
```

Заповнюємо першу таблицю:

```
ip route add $P1_NET dev $IF1 src $IP1 table T1
```

```
ip route add default via $P1 table T1
```

Тобто ми додаємо маршрути, в яких указуємо що потрапити до підмережі першого провайдера можна через перший інтерфейс. У другому рядку ми додаємо шлюз за замовчуванням. Аналогічно до першої таблиці заповнюємо другу:

```
ip route add $P2_NET dev $IF2 src $IP2 table T2
```

```
ip route add default via $P2 table T2
```

Далі опрацьовуємо головну таблицю під назвою «main». Її ми бачимо, коли вводимо ip route:

```
ip route add $P1_NET dev $IF1 src $IP1
```

```
ip route add $P2_NET dev $IF2 src $IP2
```

```
ip route add default via $P1 metric 10
```

Перші два рядки аналогічні попереднім за винятком «table main». В третьому рядку вказаний маршрут за замовчуванням та метрика. На цьому робота з маршрутизацією закінчилася. Для виводу інформації про правила

					ВКРБ-123.23.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

потрібно набрати «ip route show table <ім'я таблиці>». Тепер можна долучити правила завдяки яким і буде визначено який пакет піде через яку таблицю маршрутизації:

```
ip rule add from $IP1 table T1
```

```
ip rule add from $IP2 table T2
```

Тут ми визначаємо: якщо адреса входу дорівнює першій адресі, то маршрутизація виконується по першій таблиці T1. Аналогічно другий запис.

```
ip rule add from $SRV11 fwmark 10 table T1
```

```
ip rule add from $SRV12 fwmark 20 table T2
```

Використання iptables дозволяє маркувати пакети та маршрутизувати їх на основі даних маркерів. Отже тут ми добавили два правила для пакетів з мітками 10 та 20 відповідно для таблиць T1 та T2. Отже ми регламентували використання одночасно обох Інтернет каналів. Тепер напишемо правила які будуть маршрутизувати наші мережі та користувачів.

Доцільно використовувати форму виконавчого файлу. Це дає змогу вносити корективи не зупиняючи процес. Розглянемо головні блоки правил які використовуються в нашій системі.

Змінні для позначення назв інтерфейсів та IP адрес:

1-й канал Інтернету;

```
CH1=XXX.XXX.XXX.XXX
```

2-й канал Інтернету;

```
CH2=YYY.YYY.YYY.YYY
```

Демілітаризована мережа;

```
WinIP=192.168.50.1
```

Локальна мережа;

```
LinIP=192.168.0.1
```

Віддалена мережа;

```
RinIP=192.168.2.1
```

Інтерфейс 1-го Інтернет каналу;

					<b>ВКРБ-123.23.0004.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45











- Визначаємо мережеві налаштування демілітаризованої мережі:  
`dhcp-range=green:192.168.50.10,192.168.0.199,255.255.255.0,72h`  
`dhcp-option=green:option:router,192.168.50.1`  
`dhcp-option=green:3,192.168.50.1`  
`dhcp-option=green:6,192.168.50.1`  
`dhcp-option=green:option:ntp-server,192.168.0.130`
- Визначаємо адреси серверів аунтифікації та термінального серверу:  
`address=/vwsrver/192.168.0.240`  
`address=/tmsrver/192.168.0.250`
- Активуємо логування підключених клієнтів:  
`dhcp-leasefile=/var/log/dnsmasq.leases`
- Активуємо та визначаємо налаштування для сервісних програм які запускаються до завантаження комп'ютера і слугують для обслуговування:  
`enable-tftp`  
`tftp-root=/var/lib/tftpboot`  
`dhcp-boot=pxelinux.1,boothost,192.168.0.254`  
`pxe-prompt="Press F8 for PXE Network boot.", 2`  
`pxe-service=x86PC, "Boot via PXE",pxelinux`

Таким чином, завдання визначене ТЗ та постановкою задачі, виконане в повному обсязі. Основні етапи розробки ПЗ підсистеми включають:

- розробку структурної і функціональної схем підсистеми для визначення та обґрунтування вибору структурних частин програмної моделі ті їх функцій;
- розробку підсистеми відповідно принципам та методиці реалізації систем аналогічного класу;
- теоретичне обґрунтування розробки основних рішень побудови програмної моделі та виконання необхідних розрахунків;
- розробка алгоритмів та ПЗ підсистеми.



- має прийнятну вартість для мінімальної конфігурації комплексу технічних засобів і можливість використання обладнання, яке відповідає сучасним технічним досягненням;

- підтримує механізми транзакцій та реплікацій при багатокористувацькій роботі; забезпечує сумісність ПЗ з іншими апаратними платформами.

#### 4.2 Захист розробленого програмного забезпечення

Стрімкий розвиток програмування – невід’ємна частина сучасного рівня розвитку технологій. ПЗ давно сформувалось як конкурентноспроможний повноцінний продукт, що зазвичай же призвело до створення ринку, на якому можна продати чи придбати комп’ютерні програми. Саме тому виникла необхідність регулювати відносини між правовласниками програмних продуктів.

Згідно закону України від 23.12.93 р. № 3792-ХІІ “Про авторське право і суміжні права” в редакції від 16.07.01 року визначено терміни “комп’ютерна програма” та “база даних”. І звичайно ж, згідно цього ж Закону, комп’ютерна програма та БД є об’єктами авторського права, тобто інтелектуальною власністю.

Основною проблемою, яка пов’язана зі специфікацією комп’ютерної програми як об’єкта авторського права, на даний час є нелегальне розповсюдження та використання “піратських копій” ПЗ, що порушує права власників на ПЗ. Саме тому необхідно створити повноцінний захист інтелектуальної власності, особливо, зважаючи на катастрофічне положення в Україні, пов’язане з нелегальним розповсюдженням копій ПЗ.

На даний час існує багато способів захисту ПЗ від незаконного розповсюдження та використання. Вони поділяються на програмні, організаційні, технічні та юридичні.

					<b>ВКРБ-123.23.0004.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

На самому ефективному етапі боротьби з порушеннями – етапі профілактики – реальний захист виключних прав на комп'ютерні програми можна забезпечити лише встановленням засобів програмного захисту. Задача розробника ПЗ полягає в тому, щоб користувач, встановивши задачу на ПЕОМ, не зміг би використати її внаслідок того, що спрацюють засоби програмного захисту від нелегального копіювання. Це – завдання програмістів, яке має бути реалізоване на етапі розробки ПЗ, або на початку промислової експлуатації системи на підприємстві (в організації).

Багато програмних продуктів зараз створюються з використанням механізму активації (наприклад, продукти Microsoft), згідно яких встановлену програму необхідно активізувати через Internet або іншими, вказаними авторами, шляхами, і тільки після цього можливе її подальше використання та робота з нею.

Прикладом технічного захисту є можливість використання комплексу технічних засобів для захисту ПЗ. Це так звані апаратні ключі, які є способами захисту ПЗ від незаконного тиражування або його установку на більшу кількість ПЕОМ, ніж дозволено. Технологічно може бути виконаним у вигляді невеликого пристрою, який приєднується до одного з портів ПЕОМ. Кожний ключ дозволяє використовувати тільки одну з встановлених версій програми.

Організаційні заходи – це створення спеціальних груп в організаціях - розробниках ПЗ, які займаються політикою ліцензування, спрямованою на забезпечення правомірного використання ПЗ.

Юридичні аспекти захисту інтелектуальної власності закріплені в законодавстві України, додержання їх необхідне для формування цивілізованого ринку програмного продукту в країні. Проте, щоб виявити порушення по незаконному використанню програмного продукту, необхідні наявність розвиненого апарату для відстеження цього питання.

					<b>ВКРБ-123.23.0004.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54



- свободу удосконалення програми й випуску покращеної версії у публічний доступ (попередньою умовою цього є гарантований доступ до початкового коду).

Тому захист системи реалізується за допомогою аутентифікації. Перед початком роботи з системою користувач вводить свій логін та пароль в разі їх відповідності користувач одержує дозвіл на роботу з системою. Якщо пароль/логін не співпадає з еталоном, то програма чекає введення правильного логіна та паролю. Тільки користувач з привілеями адміністратора мережі може створити обліковий запис користувача та надати потрібні права.

					ВКРБ-123.23.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

## 5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Згідно з технічним завданням в процесі роботи над бакалаврською роботою було розроблене апаратно-програмне забезпечення для системного адміністрування та управління неоднорідною мережею на базі Soft Linux рішень. Визначена архітектура комплексу технічних засобів, що забезпечать функціонування підсистеми.

Розроблена підсистема може бути впроваджена в промислову експлуатацію в будь-якій сфері діяльності людини при використанні автоматизованої обробки даних на ПК та в сферах які передбачають використання мереж взагалі.

Підсистема, що була розроблена в процесі виконання бакалаврської роботи, являє собою комплекс апаратних та програмних засобів, структура яких та зв'язок між детально описані в розділах 3 та 4 пояснювальної записки, і забезпечує вирішення актуальних в наш час задач, тому в даному розділі розглядаються загальні заходи по впровадженню підсистеми в промислову експлуатацію.

Впровадження підсистеми може бути ефективно лише при умові наявності методичного (МЕТОЗ) та організаційного (ОРЗ) забезпечень системи.

МЕТОЗ – це комплекс організаційно-технічних заходів, направлених на забезпечення достовірності інформації про організацію впровадження та промислової експлуатації підсистеми.

МЕТОЗ на стадії впровадження підсистеми в промислову експлуатацію – це комплекс документів, що описують і утримують правила та інструкції по використанню програмного забезпечення, а саме:

- інструкції по використанню програмного забезпечення;
- інструкції користувача, який буде працювати з підсистемою;

- правила та інструкції по забезпеченню виконання вимог ТБ та промсанітарії при експлуатації підсистеми;
- таблиці нормативної довідкової інформації;
- журнали для реєстрації роботи користувачів, обліку машинного часу, обліку проведених регламентних та ремонтних робіт та інші;
- штатний розклад обслуговуючого персоналу;
- посадові інструкції обслуговуючого персоналу;
- інструкції по супроводженню промислової експлуатації програмного забезпечення;
- заходи щодо впровадження передових методів праці, планів наукової організації праці, принципів морального та матеріального стимулювання робітників.

Не менш важливою складовою успішного промислового функціонування підсистеми є організаційне забезпечення (ОРЗ). ОРЗ – це набір відповідних правил, нормативних і юридичних документів, нормативних актів, що забезпечують та регламентують процес утворення, експлуатації і розвитку підсистеми в будь-якій організації. Сюди можна віднести й відповідні правила експлуатації та обслуговування тих чи інших технічних засобів, тобто організацію і час профілактики, періодичні тести, тощо.

При розробці підсистеми необхідно керуватись положеннями концепції побудови систем аналогічного спрямування та класу, діючими нормативними та методичними документами, наказами відповідних керівників: підприємств, підрозділів, що будуть експлуатувати ПЗ.

Обов'язкові заходи по створенню системи:

- розробка ТЗ для проектування та його затвердження відповідними керівниками;
- виконання технічного проекту на основі затвердженого ТЗ, узгодження та затвердження проекту;
- розробка робочої документації по монтажу і наладці;

					<b>ВКРБ-123.23.0004.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58



- терміни розробки проектної документації по елементах підсистеми;
- терміни постачання технічних засобів та їх монтажу;
- терміни виконання пусконаладжувальних робіт.

У плані впровадження зазначаються організації (підрозділи), які відповідають за виконання робіт. Тому для впровадження розробленої програми в експлуатацію потрібна взаємодія програміста-розробника з користувачем, ІТР по обслуговуванню технічного обладнання, керівників різних рівнів, підрозділів, які будуть експлуатувати цю систему.

Кафедра \_ КБПЗ \_ 2023 рік

					<b>ВКРБ-123.23.0004.00.00.ПЗ</b>	Арк.
<i>Вим.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<b>60</b>

## 6. ОСНОВНІ ВИСНОВКИ

Згідно ТЗ в процесі виконання бакалаврської роботи розроблене апаратно-програмне забезпечення побудови неоднорідної мережі на базі Soft Linux рішень виконаний наступний обсяг запланованої роботи:

1) у вступі проведено ознайомлення з темою, що підлягала розробці в процесі виконання бакалаврської роботи. Визначена і обґрунтована актуальність теми БР, яка полягає в наступному: фактично системи аналогічного спрямування та класу на ринку програмних продуктів відсутні. Тому доцільно виконати розробку, заплановану темою БР;

2) у розділі 1 пояснювальної записки приводяться: основні фактори, що визначають необхідність розробки; специфічні терміни (надається їх пояснення), параметри підсистеми, що підлягає розробці. Детально визначена сутність майбутньої підсистеми, визначена стандартна структура підсистеми та означені її функціональні можливості. Визначено призначення розробленої системи. Окрім цього, розроблена підсистема може бути використана у навчальних закладах будь-якого рівня акредитації в якості своєрідного тренажера для підвищення ефективності вивчення курсу дисциплін з основ комп'ютерної та системної інженерії та набуття студентами практичних навичок по роботі з системами аналогічного класу та спрямування;

3) у розділі 2 пояснювальної записки приведені стислі характеристики існуючих систем-аналогів та програм-аналогів: Online Armor Personal Firewall Free, Outpost Firewall Free, ZoneAlarm Free, PC Tools Firewall Plus, Comodo Firewall Pro, Kerio Control, Jetico Personal Firewall, Sunbelt Kerio Personal Firewall, Samba, Nmap та надані висновки, на основі проведеного аналізу їх властивостей, щодо виявлених недоліків та позитивних сторін. На підставі проведеного аналізу робиться висновок про доцільність розробки підсистеми згідно теми БР. Визначені та обґрунтовані основні принципи та методика проектування майбутньої системи. Виконано обґрунтування вибору, в якості

					<b>ВКРБ-123.23.0004.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61



- список літератури, що використовувалась при розробці ПЗ та написанні пояснювальної записки.

Для більш детального пояснення текстовий матеріал ілюструється графічним матеріалом, який як входить в склад самих розділів пояснювальної записки, так і додається до пояснювальної записки в кількості 4 листів формату А4, а саме:

- структурна схема;
- функціональна схема;
- діаграма взаємодії процесів;
- блок-схема алгоритму функціонування основних програмних модулів системи..

Проведена в рамках ТЗ дослідна експлуатація системи показала, що вона цілком дієздатна та дуже проста для сприйняття, хоча і вимагає від користувача визначеного рівня знань з питань системного програмування та роботи з ресурсами ПК та мережі.

Таким чином, ТЗ та постановка задачі на розробку ПЗ підсистеми системного адміністрування та управління неоднорідною мережею на базі Soft Linux рішень виконані в повному обсязі. Розроблене програмно-апаратне рішення забезпечує взаємодію віддалених мереж з різним призначенням. Система надає можливість об'єднати і централізувати управління систем диспетчеризації, телефонії, відеонагляду, мережі корпоративного та загального призначення, тобто - відповідає сучасним вимогам до систем аналогічного класу.

					ВКРБ-123.23.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. <http://km.ptngu.com/lections/17.html> - Комп'ютерні мережі.  
Електронний посібник
2. <http://infdis.narod.ru/adm/ais-n4.htm> - Администрирование информационных сетей.
3. <https://habr.com/ru/post/118817/> - NetPing: конструктор для администратора .
4. <http://gallery.ru-board.com/topic.cgi?forum=8&topic=2988&start=140> -  
Программы для мониторинга сети Windows
5. FreeBSD 9. Корпоративный Интернет-сервер. / Корниенко К.А. -  
К.:ФЛП Декет В.М., 2013 — 176 с.
6. Авдеев, В. А. Периферийные устройства: интерфейсы, схемотехника,  
программирование : учебное пособие для студентов вузов, обучающихся по  
специальности 230101 "Вычислительные машины, комплексы, системы и  
сети" / В. А. Авдеев. - Москва : ДМК Пресс, 2016. – 846 с.
7. <https://nmap.org/man/ru/index.html> - Справочное руководство Nmap  
(Man Page).
8. Linux. От новичка к профессионалу. - 6-е изд., перераб. и доп./  
Колисниченко Д.Н. СПб.: БХВ-Петербург, 2018. - 672 с.
9. Технологія проектування програмних систем: Методичні вказівки до  
виконання лабораторних робіт для студентів денної і заочної форми навчання  
за напрямом підготовки 7.050102.01, 8.050102.01 «Комп'ютерні системи та  
мережі» 7.050102.02, 8.050102.02 «Системне програмування» / Укл.: / Смірнов В.В.,  
Смірнова Н.В. – Кіровоград: КНТУ, 2014. – 85 с. (Електронний ресурс)
10. Штаєр Л.О. Технології розробки програмного забезпечення; конспект  
лекцій. – Івано-Франківськ: ІФНТУНГ, 2017. – 139 с. (Електронний ресурс)
11. Дрововозов В.І. Експлуатація комп'ютерних систем та мереж  
Навчальний посібник. - К.: НАУ. – 2007. – 361 с.

					ВКРБ-123.23.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64



23. Комп'ютерні мережі та телекомунікації : навчальний посібник / Азарова А. О., Лисак Н. В. – Вінниця : ВНТУ, 2012. – 293 с.

24. Денісова О. О. Інформаційні системи і технології в юридичній діяльності : навч. посіб. / Денісова О. О. – К. : КНЕУ, 2003. – 315 с.

25. Таненбаум Э., Уэзеролл Д. Компьютерные сети. 5-е изд. — СПб.: Питер, 2012. — 960 с.: ил.

Кафедра КБПЗ – 2023 рік

					ВКРБ-123.23.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

Додатки

Кафедра \_ КБПЗ \_ 2023рік

					ВКРБ-123.23.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

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 Етапи розробки.....	5
9 Порядок контролю та приймання.....	6

**ВКРБ-123.23.0004.00.00.ТЗ**

Вим.	Арк.	№ документа	Підпис	Дата				
Розробив		Козак А.І.			Програмне забезпечення підсистеми системного адміністрування та управління неоднорідною мережею	Літ.	Аркуш	Аркушів
Перевірів		Пархоменко Ю.М.				Б	1	6
Н. Контр.		Гермак В.С.			ЦНТУ КМ-19			
Затв.		Смірнов О.А.						

## **1 Найменування та область застосування**

Це технічне завдання розповсюджується на дослідження та програмну реалізацію побудови неоднорідної мережі на базі Soft Linux рішень.

## **2 Підстава для розробки**

Підставою для розробки служить завдання на бакалаврську роботу, видане на кафедрі програмування комп'ютерних систем і мереж (нак. №10-02 від 5.01.2023 року).

## **3 Мета та призначення розробки**

Метою бакалаврської роботи є дослідження та програмна реалізація автоматичного виконання основних функцій та утиліт по системному адмініструванню та скануванню мереж і одночасній роботі з різними мережними ресурсами в віддалених мережах при їх об'єднанні.

## **4 Джерела розробки**

Джерелом цієї бакалаврської роботи 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 системи, а також розробка взаємодії враховуючи використання декількох каналів Інтернет;

					<b>ВКРБ-123.23.0004.00.00.ТЗ</b>	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

– розробка програми, що реалізує спроектовані алгоритми роботи системи.

## 5.2 Показники призначення

Система повинна забезпечувати:

- одночасне використання декількох каналів Інтернету;
- об'єднання віддалених мереж;
- сканування ресурсів мережі з вільним доступом;
- виведення різнобічної інформації про мережні пристрої.

## 5.3 Вимоги до функціональних характеристик

Розроблена система повинна задовольняти усі критерії маршрутизації та мати можливість масштабування.

## 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

Система повинна відповідати усім критеріям та стандартам, які притаманні аналогічним системам та мати високу надійність при експлуатації.

## 5.6 Умови експлуатації

Місце дислокації повинно задовольняти наступним умовам експлуатації:

					<b>ВКРБ-123.23.0004.00.00.ТЗ</b>	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

- 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 Linux 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 Linux

#### 5.8.1 Обладнання

Комп'ютер Intel® Core Duo/2048 Mb/80 Gb/SVGA 14" 1Mb або сумісні з ним.

#### 5.8.2 Мова програмування

В системі використовується мова програмування C та синтаксис написання bash скриптів.

#### 5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

					<b>ВКРБ-123.23.0004.00.00.ТЗ</b>	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		4

#### 5.8.4 Вихідні дані

Конфігураційні файли.

#### 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 Перелік документів, що розробляються

– Структурна схема системи	– 1
аркуш.	
– Функціональна схема системи	
– 1 аркуш.	
– Діаграма процесів	
– 1 аркуш.	
– Блок-схема алгоритму роботи програми	– 2
аркуші.	
– Пояснювальна записка	– 75
аркушів.	

#### 8 Етапи розробки

8.1 Збір і обробка інформації по темі бакалаврської роботи. Постановка задачі на виконання бакалаврської роботи (складання ТЗ).

					<b>ВКРБ-123.23.0004.00.00.ТЗ</b>	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

8.2 Проведення досліджень або експериментальних робіт для уточнення основних положень бакалаврської роботи.

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

8.4 Побудова схем взаємодії даних.

8.5 Створення прототипу ПЗ.

8.6 Віднаходження ПЗ, аналіз отриманих результатів.

8.7 Оформлення пояснювальної записки і виконання робіт по графічній частині.

## 9 Порядок контролю та приймання

9.1 Подання бакалаврської роботи на попередній захист 16.05.2023 р.

9.2 Подання бакалаврської роботи на захист 21.05.2023 р.

					ВКРБ-123.23.0004.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б  
(обов'язковий)

**Міністерство освіти і науки України**  
**Центральноукраїнський національний технічний університет**

ЗАТВЕРДЖУЮ

Керівник бакалаврської роботи

\_\_\_\_\_ Пархоменко Ю.М.

**Програмне забезпечення підсистеми системного  
адміністрування та управління неоднорідною мережею**

Лістинг програми

Код документу 12

Носій: CD-R диск

Загальна кількість аркушів: 17

Літера: РП

Кропивницький 2023

***/etc/iproute2/rt\_tables***

```

IF1=eth0
IF2=eth1
//IF – Мережеві інтерфейси двох провайдерів.
IP1=XXX.XXX.XXX.XXX
IP2=YYY.YYY.YYY.YYY
//IP – зовнішні IP-адреси, котрі видають провайдери.
P1=XXX.XXX.XXX.XXX
P2=YYY.YYY.YYY.YYY
//P – шлюзи від провайдерів.
SRV11=192.168.0.254
SRV12=192.168.0.253
SRV11 та SRV12

echo «101 T1» >> /etc/iproute2/rt_tables
echo «102 T2» >> /etc/iproute2/rt_tables

ip route add $P1_NET dev $IF1 src $IP1 table T1
ip route add default via $P1 table T1

ip route add $P2_NET dev $IF2 src $IP2 table T2
ip route add default via $P2 table T2

ip route add $P1_NET dev $IF1 src $IP1
ip route add $P2_NET dev $IF2 src $IP2
ip route add default via $P1 metric 10

ip rule add from $IP1 table T1
ip rule add from $IP2 table T2

ip rule add from $SRV11 fwmark 10 table T1
ip rule add from $SRV12 fwmark 20 table T2

```

***route.sh***

```

//1-й канал Інтернету;
CH1=XXX.XXX.XXX.XXX
//2-й канал Інтернету;
CH2=YYY.YYY.YYY.YYY
// Демілітаризована мережа;
WinIP=192.168.50.1
//Локальна мережа;

```

```

LinIP=192.168.0.1
//Віддалена мережа;
RinIP=192.168.2.1
//Інтерфейс 1-го Інтернет каналу;
CH1inL=eth0
//Інтерфейс 2-го Інтернет каналу;
CH2inL=eth1
//Інтерфейс WiFi;
WinL=eth2
//Інтерфейс локальної мережі;
LinL=eth3

//Видаляємо правила та чистимо таблиці;
iptables -t filter -F
iptables -t filter -X
iptables -t mangle -F
iptables -t mangle -X
iptables -t nat -F
iptables -t nat -X
//Записуємо нові правила:
//Дозволяємо вхід пакетів на кільцевий інтерфейс:
iptables -A INPUT -i lo - ACCEPT
//Дозволяємо вхід пакетів на кільцевий інтерфейс на який повертає
пакети другий з //провайдерів:
iptables -A INPUT -i lo:0 -j ACCEPT
//Дозволяємо пінг з обмеженням по вартості пакетів:
iptables -A INPUT -p icmp --icmp-type 0 -j ACCEPT
iptables -A INPUT -p icmp --icmp-type 3 -j ACCEPT
iptables -A INPUT -p icmp --icmp-type 8 -j ACCEPT
iptables -A INPUT -p icmp --icmp-type 11 -j ACCEPT
iptables -A INPUT -s 192.168.0.0/24 -p TCP --dport 25 -j ACCEPT
//Дозволяємо вхід та вихід для пошти:
iptables -A INPUT -s 192.168.0.0/24 -p TCP --dport 110 -j ACCEPT
iptables -A INPUT -s 192.168.0.0/24 -p TCP --dport 465 -j ACCEPT
iptables -A INPUT -s 192.168.0.0/24 -p TCP --dport 995 -j ACCEPT
//Дозволяємо обмін пакетів у мережах:
iptables -A INPUT -i $LinL -j ACCEPT
iptables -A INPUT -i $WinL -j ACCEPT

iptables -A INPUT -i $CH1inL -j ACCEPT
iptables -A INPUT -i $CH2inL -j ACCEPT
iptables -A INPUT -i $RinIP -j ACCEPT

```

```

//Дозволяємо DNS трафік:
iptables -A INPUT -s 192.168.0.0/24 -p UDP --dport 5353 -j ACCEPT
iptables -A INPUT -s 192.168.50.0/24 -p UDP --dport 5353 -j ACCEPT
iptables -A INPUT -p UDP --dport 53 -j ACCEPT
iptables -A INPUT -p UDP --dport 5353 -j ACCEPT
//Дозволяема кешуючий веб сервер:
iptables -A INPUT -s 192.168.0.0/24 -i enp1s0 -p TCP --dport 3128 -j
ACCEPT
//Дозволяємо PPTP:
iptables -A INPUT -p TCP --dport 1723 -j ACCEPT
iptables -A INPUT -p GRE -j ACCEPT
//Дозволяємо та захищаємо SSH з'єднання та тунелювання:
iptables -A INPUT -p tcp --dport 2222 -m state --state NEW -m recent -
-set --name SSH
iptables -A INPUT -p tcp --dport 2222 -m state --state NEW -m recent -
-update --seconds 180 --hitcount 5 --name SSH -j REJECT --reject-with icmp-
port-unreachable
iptables -A INPUT -p tcp --dport 2222 -j ACCEPT
//Демілітаризуємо WiFi мережу:
iptables -A FORWARD -i $WinL -o $LinL -j DROP
//Дозволяємо IP, не використовуючи кешуючий проксі:
iptables -A FORWARD -s 192.168.0.230 -j ACCEPT
iptables -A FORWARD -d 192.168.0.230 -j ACCEPT
//Пропускаємо через мережу доступ до приладів та додаємо віддалену
мережу:
--route add -net 192.168.2.0 netmask 255.255.255.0 gw 192.168.0.254
dev eth1
iptables -t nat -A PREROUTING -p tcp -d $RinIP --dport 5555 -j DNAT --
to-destination 192.168.2.5:5555
iptables -t nat -A PREROUTING -p tcp -d $RinIP --dport 5556 -j DNAT --
to-destination 192.168.2.6:5555
iptables -t nat -A PREROUTING -p tcp -d $RinIP --dport 5557 -j DNAT --
to-destination 192.168.2.7:5555
iptables -t nat -A PREROUTING -p tcp -d $RinIP --dport 5559 -j DNAT --
to-destination 192.168.2.9:5555
//Пропускаємо через мережу доступ до відео нагляду:
iptables -t nat -A PREROUTING -p tcp -d $ChinIP --dport 37777 -j DNAT
--to-destination 192.168.0.249:37777
iptables -t nat -A PREROUTING -p tcp -d $ChinIP --dport 37778 -j DNAT
--to-destination 192.168.0.249:37778
//Пропускаємо через мережу доступ до поштового серверу:

```

```
iptables -t nat -A PREROUTING -p tcp -d $CHinIP --dport 443-j DNAT --  
to-destination 192.168.0.249:443
```

```
iptables -t nat -A PREROUTING -p tcp -d $CHinIP --dport 80 -j DNAT --  
to-destination 192.168.0.249:80
```

//Дозволяємо користувачам WiFi мережі користуватись Інтернетом:

```
iptables -A FORWARD -s 192.168.50.0/24 -j ACCEPT
```

```
iptables -A FORWARD -d 192.168.50.0/24 -j ACCEPT
```

//Забороняємо будь-який обмін пакетів з локальною мережею:

```
iptables -A FORWARD -s 192.168.50.0/24 -d 192.168.0.0/24 -j DROP
```

```
iptables -A FORWARD -d 192.168.50.0/24 -s 192.168.0.0/24 -j DROP
```

//Забороняємо будь-який обмін пакетів з віддаленою мережею:

```
iptables -A FORWARD -s 192.168.50.0/24 -d 192.168.2.0/24 -j DROP
```

```
iptables -A FORWARD -d 192.168.50.0/24 -s 192.168.2.0/24 -j DROP
```

//Дозволяємо віддаленій мережі користуватись Інтернетом:

```
iptables -A FORWARD -s 192.168.2.0/24 -j ACCEPT
```

```
iptables -A FORWARD -d 192.168.2.0/24 -j ACCEPT
```

//Деякі приклади дозволів для користувачів не використовуючи

кешуючий проксі сервер:

```
iptables -A FORWARD -s 192.168.0.1/24 -j ACCEPT
```

```
iptables -A FORWARD -d 192.168.0.1/24 -j ACCEPT
```

```
iptables -A FORWARD -s 192.168.0.2/24 -j ACCEPT
```

```
iptables -A FORWARD -d 192.168.0.2/24 -j ACCEPT
```

```
iptables -A FORWARD -s 192.168.0.3/24 -j ACCEPT
```

```
iptables -A FORWARD -d 192.168.0.3/24 -j ACCEPT
```

```
iptables -A FORWARD -s 192.168.0.4/24 -j ACCEPT
```

```
iptables -A FORWARD -d 192.168.0.4/24 -j ACCEPT
```

```
iptables -A FORWARD -s 192.168.0.5/24 -j ACCEPT
```

```
iptables -A FORWARD -d 192.168.0.5/24 -j ACCEPT
```

```
iptables -A FORWARD -s 192.168.0.6/24 -j ACCEPT
```

```
iptables -A FORWARD -d 192.168.0.6/24 -j ACCEPT
```

```
iptables -A FORWARD -s 192.168.0.7/24 -j ACCEPT
```

```
iptables -A FORWARD -d 192.168.0.7/24 -j ACCEPT
```

```
iptables -A FORWARD -s 192.168.0.8/24 -j ACCEPT
```

```
iptables -A FORWARD -d 192.168.0.8/24 -j ACCEPT
```

```
iptables -A FORWARD -s 192.168.0.9/24 -j ACCEPT
```

```
iptables -A FORWARD -d 192.168.0.9/24 -j ACCEPT
```

```
iptables -A FORWARD -s 192.168.0.10/24 -j ACCEPT
```

```
iptables -A FORWARD -d 192.168.0.10/24 -j ACCEPT
```

```
iptables -A FORWARD -s 192.168.0.11/24 -j ACCEPT
```

```
iptables -A FORWARD -d 192.168.0.11/24 -j ACCEPT
```

```
iptables -A FORWARD -s 192.168.0.12/24 -j ACCEPT
```

```
iptables -A FORWARD -d 192.168.0.12/24 -j ACCEPT
```

```
iptables -A FORWARD -s 192.168.0.13/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.13/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.14/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.14/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.15/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.15/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.16/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.16/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.17/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.17/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.18/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.18/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.19/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.19/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.20/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.20/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.21/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.21/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.22/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.22/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.23/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.23/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.24/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.24/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.25/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.25/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.26/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.26/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.27/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.27/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.28/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.28/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.29/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.29/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.30/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.30/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.31/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.31/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.32/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.32/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.33/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.33/24 -j ACCEPT
```

```
iptables -A FORWARD -s 192.168.0.34/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.34/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.35/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.35/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.36/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.36/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.37/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.37/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.38/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.38/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.39/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.39/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.40/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.40/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.41/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.41/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.42/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.42/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.43/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.43/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.44/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.44/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.45/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.45/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.46/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.46/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.47/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.47/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.48/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.48/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.49/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.49/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.50/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.50/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.51/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.51/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.52/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.52/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.53/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.53/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.54/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.54/24 -j ACCEPT
```

```

iptables -A FORWARD -s 192.168.0.55/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.55/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.56/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.56/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.57/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.57/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.58/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.58/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.59/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.59/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.60/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.60/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.240/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.240/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.250/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.250/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.249/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.249/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.245/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.245/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.254/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.254/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.230/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.230/24 -j ACCEPT
iptables -A FORWARD -s 192.168.0.200/24 -j ACCEPT
iptables -A FORWARD -d 192.168.0.200/24 -j ACCEPT

//Маскуємо та маркуємо пакети для першого Інтернет каналу :
iptables -t nat -A POSTROUTING -s 192.168.0.0/24 -o $RinL -j SNAT --to
$CH1inIP
iptables -t nat -A POSTROUTING -s 192.168.50.0/24 -o $WinL -j SNAT --
to $CH1inIP
iptables -t nat -A POSTROUTING -s 192.168.2.0/24 -o $LinL -j SNAT --to
$CH1inIP

//Маскуємо та маркуємо пакети для другого Інтернет каналу:
iptables -t nat -A POSTROUTING -s 192.168.2.0/24 -o $RinL -j SNAT --to
$CH2inIP
iptables -t nat -A POSTROUTING -s 192.168.0.0/24 -o $WinL -j SNAT --to
$CH2inIP
iptables -t nat -A POSTROUTING -s 192.168.50.0/24 -o $LinL -j SNAT --
to $CH2inIP

```

```

//Визначаємо основні режими мережі за протоколами окрім тих, що
визначили до цього:
iptables -A INPUT -p TCP -m state --state ESTABLISHED,RELATED -j
АССЕРТ
iptables -A INPUT -p UDP -m state --state ESTABLISHED,RELATED -j
АССЕРТ

iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT ACCEPT

                                /etc/dnsmasq.conf

//Активуємо ДНС сервер:
port=53
//Визначаємо мережеві інтерфейси на яких буде працювати ДНСР:
interface=eth2
interface=eth3
//Виключаємо керування дозволами:
no-resolv
//Вказуємо вторинні сервери імен для кешуючого серверу:
server=8.8.8.8
server=8.8.4.4
//Визначаємо мережеві налаштування корпоративної мережі:
dhcp-range=red:192.168.0.60,192.168.0.199,255.255.255.0,72h
dhcp-option=red:option:router,192.168.0.254
dhcp-option=red:3,192.168.0.254
dhcp-option=red:6,192.168.0.240,192.168.0.250
dhcp-option=red:option:ntp-server,192.168.0.130
//Визначаємо мережеві налаштування демілітаризованої мережі:
dhcp-range=green:192.168.50.10,192.168.0.199,255.255.255.0,72h
dhcp-option=green:option:router,192.168.50.1
dhcp-option=green:3,192.168.50.1
dhcp-option=green:6,192.168.50.1
dhcp-option=green:option:ntp-server,192.168.0.130
//Визначаємо адреси серверів аутентифікації та термінального серверу:
address=/vwserver/192.168.0.240
address=/tmserver/192.168.0.250
//Активуємо логування підключених клієнтів:
dhcp-leasefile=/var/log/dnsmasq.leases
//Активуємо та визначаємо налаштування для сервісних програм які
запускаються до завантаження комп'ютера і слугують для обслуговування:
enable-tftp
tftp-root=/var/lib/tftpboot
dhcp-boot=pxelinux.1,boothost,192.168.0.254

```

```
pxe-prompt="Press F8 for PXE Network boot.", 2
pxe-service=x86PC, "Boot via PXE",pxelinux
```

*/керування профілями мережевих підключень/*

```
using System;
using System.Diagnostics;
using System.Security.Principal;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Diagnostics;
using System.Net;
using System.Net.NetworkInformation;
namespace NetManul
{
    public partial class Form1 : Form
    {
        private const double timerUpdate = 1000; // Timer Update
        (every 1 sec)
        private NetworkInterface[] nicArr; // Interface Storage
        private string[] notifyarr;
        private const string website =
"http://pashinskikh.github.io/";
        private const string github =
"https://github.com/pashinskikh/netmanul/";
        private const string speedtest = "http://beta.speedtest.net/";
        private const string rickroll =
"https://www.youtube.com/watch?v=dQw4w9WgXcQ";
        // Check is app running from admin
        private static bool IsAdministrator()
        {
            return (new
WindowsPrincipal(WindowsIdentity.GetCurrent())) .IsInRole(WindowsBuiltInRole.Administrator);
        }
        private void ChkNetwork()
        {
```

```

        Notify(false, "Checking internet connection...");
        if
(System.Net.NetworkInformation.NetworkInterface.GetIsNetworkAvailable() ==
true)
        {
            Notify(false, "Connected to internet!");
        }
        else
        {
            Notify(true, "No internet connection found!");
        }
    }
public Form1()
{
    InitializeComponent();
    label9.Text = "Version: " + Application.ProductVersion; //
Get app version
    if (IsAdministrator() == true)
    {
        this.Text = this.Text + " (Administrator)"; // Update
form title
    }
    InitializeNetworkInterface();
    InitializeTimer();
    ChkNetwork();
}
private void radioButton1_CheckedChanged(object sender,
EventArgs e)
{
    groupBox1.Enabled = false;
    groupBox2.Enabled = false;
    SetDefaultButton.Enabled = false;
}
private void radioButton2_CheckedChanged(object sender,
EventArgs e)
{
    groupBox1.Enabled = true;
    groupBox2.Enabled = true;
    SetDefaultButton.Enabled = true;
}
private void Notify(bool iserr, string msg = "No messages...")
{

```

```

        if (iserr == true)
        {
            ErrorLabel.ForeColor = Color.Red; // Set message color
to red (error)
        }
        else
        {
            ErrorLabel.ForeColor = Color.Black; // Set message
color to black
        }
        ErrorLabel.Text = msg; // Show notification
        NotifyTimer.Enabled = true; // Enable Timer for 3s., then
clear notification bar
    }
    private void RunCmd(string cmd)
    {
        System.Diagnostics.ProcessStartInfo startInfo = new
System.Diagnostics.ProcessStartInfo();
        startInfo.FileName =
Environment.ExpandEnvironmentVariables("%SystemRoot%") +
@"\System32\cmd.exe";
        startInfo.Arguments = "/c" + cmd;
        startInfo.WindowStyle =
System.Diagnostics.ProcessWindowStyle.Hidden;
        System.Diagnostics.Process.Start(startInfo);
    }
    // Check IP's and DNS's
    private bool ChkData()
    {
        System.Net.IPAddress myAddress;
        if (System.Net.IPAddress.TryParse(IPAddress.Text, out
myAddress) && System.Net.IPAddress.TryParse(SubnetMask.Text, out myAddress)
&& System.Net.IPAddress.TryParse(Gateway.Text, out myAddress) &&
System.Net.IPAddress.TryParse(PrefDNS.Text, out myAddress) &&
System.Net.IPAddress.TryParse(AltDNS.Text, out myAddress))
        {
            if (PrefDNS.Text == AltDNS.Text)
            {
                // DNS addresses is the same
                Notify(true, "Error: DNS addresses can\'t be the
same!");
            }
            return false;
        }
    }
}

```

```

        }
        // IP is valid
        Notify(false);
        return true;
    }
    // IP is invalid
    // MessageBox.Show("Invalid IPv4 Address!", "Error",
    MessageBoxButtons.OK, MessageBoxIcon.Error);
    Notify(true, "Error: invalid IPv4 address!");
    return false;
}
private void ChangeIPButton_Click(object sender, EventArgs e)
{
    if (radioButton1.Checked == true)
    {
        // Use DHCP
        RunCmd("netsh interface ipv4 set address \"" +
        cmbInterface.GetItemText(cmbInterface.SelectedItem) + "\" dhcp && netsh
        interface ipv4 set dns \"" +
        cmbInterface.GetItemText(cmbInterface.SelectedItem) + "\" dhcp");
    }
    if (radioButton2.Checked == true)
    {
        if (ChkData() == true)
        {
            // If IP is valid - set new settings
            RunCmd("netsh interface ipv4 set address \"" +
            cmbInterface.GetItemText(cmbInterface.SelectedItem) + "\" static " +
            IPAddress.Text + " " + SubnetMask.Text + " " + Gateway.Text + " && " + "netsh
            interface ipv4 add dns \"" +
            cmbInterface.GetItemText(cmbInterface.SelectedItem) + "\" " + PrefDNS.Text +
            " && " + "netsh interface ipv4 add dns \"" +
            cmbInterface.GetItemText(cmbInterface.SelectedItem) + "\" " + AltDNS.Text + "
            INDEX = 2");
        }
    }
}
private string Ping()
{
    try
    {
        Ping pingSender = new Ping();
    }
}

```

```

        IPAddress address =
System.Net.IPAddress.Parse(ipforservice.Text);
        PingReply reply = pingSender.Send(address);
        return "Reply from " + reply.Address.ToString() + ":"
+ " bytes=" + reply.Buffer.Length + " time=" + reply.RoundtripTime + " TTL="
+ reply.Options.Ttl;
    }
    catch
    {
        return "Destination host unreachable";
    }
}
// Open website
private void linkLabel1_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
{
    System.Diagnostics.Process.Start(website);
}
// Open Github
private void linkLabel2_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
{
    System.Diagnostics.Process.Start(github);
}
// Never Gonna Give You Up
private void pictureBox1_Click(object sender, EventArgs e)
{
    System.Diagnostics.Process.Start(rickroll);
}
// Run Ookla Speedtest
private void button1_Click(object sender, EventArgs e)
{
    System.Diagnostics.Process.Start(speedtest);
}
// Set default settings
private void SetDefaultButton_Click(object sender, EventArgs
e)
{
    IPAddress.Text = "192.168.0.101";
    SubnetMask.Text = "255.255.255.0";
    Gateway.Text = "192.168.0.1";
    PrefDNS.Text = "8.8.8.8";
}

```

```

        AltDNS.Text = "8.8.4.4";
    }
    private void InitializeNetworkInterface()
    {
        // Grab all local interfaces to this computer
        nicArr = NetworkInterface.GetAllNetworkInterfaces();
        // Add each interface name to the combo box
        for (int i = 0; i < nicArr.Length; i++)
        {
            cmbInterface.Items.Add(nicArr[i].Name);
        }

        // Change the initial selection to the first interface
        cmbInterface.SelectedIndex = 0;
    }
    private void InitializeTimer()
    {
        timer = new Timer();
        timer.Interval = (int)timerUpdate;
        timer.Tick += new EventHandler(timer_Tick);
        timer.Start();
    }
    // Update GUI components for the network interfaces
    private void UpdateNetworkInterface()
    {
        // Grab NetworkInterface object that describes the current
interface
        NetworkInterface nic = nicArr[cmbInterface.SelectedIndex];
        // Grab the stats for that interface
        IPv4InterfaceStatistics interfaceStats =
        nic.GetIPv4Statistics();
        // Try to get the IPv4 interface properties.
        try
        {
            IPInterfaceProperties adapterProperties =
        nic.GetIPProperties();
            IPv4InterfaceProperties p =
        adapterProperties.GetIPv4Properties();
        }
        catch
        {

```

```

        Notify(true, "There was a problem with loading
interfaces!");
    }
    // Calculate the speed of bytes going in and out
    int bytesSentSpeed = (int)(interfaceStats.BytesSent -
double.Parse(lblBytesSent.Text)) / 1024;
    int bytesReceivedSpeed =
(int)(interfaceStats.BytesReceived - double.Parse(lblBytesReceived.Text)) /
1024;

    int netSpeed = (int)nic.Speed / 1000000;
    // Update the labels
    lblInterfaceType.Text =
nic.NetworkInterfaceType.ToString();
    lblSpeed.Text = netSpeed.ToString() + " Mbps";
    lblBytesReceived.Text =
interfaceStats.BytesReceived.ToString();
    lblBytesSent.Text = interfaceStats.BytesSent.ToString();
    lblUpload.Text = bytesSentSpeed.ToString() + " KB/s";
    lblDownload.Text = bytesReceivedSpeed.ToString() + "
KB/s";
}
// The Timer event for each Tick (second) to update the UI
void timer_Tick(object sender, EventArgs e)
{
    UpdateNetworkInterface();
}
private void NotifyTimer_Tick(object sender, EventArgs e)
{
    Notify(false);
    NotifyTimer.Enabled = false;
}
private void label9_Click(object sender, EventArgs e)
{
    Form2 AboutForm = new Form2();
    AboutForm.Show();
}
private void button2_Click(object sender, EventArgs e)
{
    System.Net.IPAddress myAddress;
    if (System.Net.IPAddress.TryParse(ipforservice.Text, out
myAddress))
    {

```

```
        tooltimer.Enabled = true;
    }
    else
    {
        Notify(true, "Incorrect IP!");
    }
}
private void tooltimer_Tick(object sender, EventArgs e)
{
    textBox2.AppendText(Ping() + "\n");
}
private void button3_Click(object sender, EventArgs e)
{
    tooltimer.Enabled = false;
}
private void button4_Click(object sender, EventArgs e)
{
    ipforservice.Text = "8.8.8.8";
    tooltimer.Enabled = false;
    textBox2.Clear();
}
}
}
```

Кафедра — КБІІБ — 2023рік