

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

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

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи передачі даних
в IRC-мережах”

КБПЗ - 2023

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

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

Центральноукраїнський національний технічний університет

Центр *Заочної та дистанційної освіти*

Кафедра *Кібербезпеки та програмного забезпечення*

Рівень вищої освіти *магістр*

Галузь знань 12 "Інформаційні технології"

Спеціальність 122 "Комп'ютерні науки"

Освітньо-професійна (освітньо-наукова) програма "Комп'ютерні науки"

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 6 » вересня 2023 року

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

Трапезніковій Наталії Вікторівні

(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження та програмна реалізація системи передачі даних в IRC-мережах*

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

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

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

3. Строк подання студентом роботи до захисту 10.12.2023 р.

4. Мета та завдання випускної кваліфікаційної роботи: *Метою розробки є дослідження та програмна реалізація системи передачі даних в IRC-мережах*

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

1. Призначення та область використання.

6. Наукова новизна.

2. Перегляд аналогічних існуючих систем.

7. Економічна ефективність розробленої програми.

3. Опис і обґрунтування проектних рішень.

8. Заходи з охорони праці та техніки безпеки.

4. Етапи програмування системи.

9. Висновки.

5. Впровадження системи в промислову експлуатацію

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

Наукова новизна

1 аркуш

Структурна схема системи

1 аркуш

Функціональна схема системи

1 аркуш

Діаграма процесів

1 аркуш

Блок-схема алгоритму роботи додатку

2 аркуша

Показники економічної ефективності

1 аркуш

6. Консультанти розділів роботи

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

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

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

Дата видачі завдання
« 6 » вересня 2023 р.

Підпис керівника

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

Підпис здобувача

(прізвище та ініціали)

АНОТАЦІЯ

Трапезнікова Н.В. Дослідження та програмна реалізація системи передачі даних в IRC-мережах. 122 Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2023.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи передачі даних в IRC-мережах.

Метою розробки є дослідження та програмна реалізація системи передачі даних в IRC-мережах.

Об'єктом дослідження є процес передачі даних в IRC-мережах.

Предметом дослідження є методи передачі даних в IRC-мережах.

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

Результат роботи – програмна реалізація системи передачі даних в IRC-мережах.

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

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

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

Програму розроблено в середовищі Delphi 10.4 Sydney.

Ключові слова: комп'ютерні науки, IRC-мережа

ABSTRACT

Trapeznikova N.V. Research and software implementation of the data transmission system in IRC networks. 122 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2023.

In this graduation thesis for the second (master's) level of higher education, software is developed, which is intended for the data transmission system in IRC networks.

The purpose of the development is research and software implementation of the data transmission system in IRC networks.

The object of research is the process of data transfer in IRC networks.

The subject of the study is data transfer methods in IRC networks.

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

The result of the work is the software implementation of the data transmission system in IRC networks.

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

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

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

The program was developed in the Delphi 10.4 Sydney environment.

Keywords: computer science, IRC network

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	6
1.1 Призначення системи.....	6
1.2 Область застосування.....	6
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	8
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	8
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	16
2.3 Розгорнута постановка завдання	22
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	23
3.1 Опис функціонування системи	23
3.2 Розробка структурної схеми.....	29
3.3 Розробка функціональної схеми	33
3.4 Розробка діаграми процесів.....	49
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	52
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	52
4.2 Захист розробленого програмного забезпечення.....	70
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	73
6 НАУКОВА НОВИЗНА	75

						ВКРМ-122.23.0070.00.00.ПЗ		
Вим	Арк.	№ докум.	Підп.	Дата				
Розроб.	Трапезнікова Н.В.				Дослідження та програмна реалізація системи передачі даних в IRC-мережах	Літ.	Аркуш	Аркушів
Перев.	Дресва Г.М.					М	1	114
Н.контр.	Коваленко А.С.				ЦНТУ КН-22МЗ			
Затв.	Смірнов О.А.							

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	76
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	76
7.2 Розрахунок трудомісткості розробки програмної продукції.....	78
7.3 Визначення чисельності виконавців і планового фонду зарплати.....	80
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника.....	85
7.5 Визначення собівартості розробки та ціни програмної продукції.....	89
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	92
7.7 Визначення експлуатаційних витрат.....	93
7.8 Визначення економічної ефективності програмної продукції.....	94
7.9 Висновок.....	96
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	97
8.1 Вступ.....	97
8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером.....	98
8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста ...	99
8.4 Розробка заходів з умов поліпшення охорони праці.....	102
8.5 Висновки до розділу.....	103
9 ОСНОВНІ ВИСНОВКИ.....	106
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	108

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

DCC	–	Direct Client-to-Client
HTTP	–	HyperText Transfer Protocol
IPv6	–	Internet Protocol version 6
SSL	–	Secure Sockets Layer

КБПЗ_2023

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

Актуальність теми. IRC – Internet Relay Chat – це протокол, що дозволяє користувачам спілкуватися один з одним у реальному часі за допомогою набору слів на клавіатурі (chat). З IRC зараз працюють тисячі користувачів Інтернет по усьому світі. IRC може служити як і для розваги, так і для цілком серйозних справ: допомоги й консультації в роботі, передачі інформації й т.п.

IRC складається із серверів. Сервер може бути з'єднаний з іншими серверами. Сукупність серверів, з'єднаних один з одним утворює мережа. У світі існують десятки мереж. Найбільш старі й відомі – це IRCNet і EfNet, що утворилася з розколеної першої IRC мережі.

Для роботи з IRC необхідна спеціальна програма клієнт. Далі треба підключитися до сервера

Після підключення до сервера можна знайти потрібну людину й спілкуватися з ним. Можна також підключитися до каналу. Спілкування на каналі нагадує спілкування людей у кімнаті. Хто-небудь говорить і всі його чують. У кожного каналу є ім'я, що відбиває загальну тематику розмови. Це може бути як тема, що цікавить всіх (наприклад, #linux, #android), так і просто яка-небудь об'єднуюча всіх назва (наприклад, #kirovohrad, #odessa). Якщо в першому випадку розмови зазвичай ведуться навколо зазначеної тематики, то в другому випадку розмови ведуться на зовсім різні теми.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи передачі даних в IRC-мережах.

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

- Огляд існуючих систем передачі даних в IRC-мережах.
- Дослідження системи передачі даних в IRC-мережах.
- Програмна реалізація системи передачі даних в IRC-мережах.

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

Об'єктом дослідження є процес передачі даних в IRC-мережах.

Предметом дослідження є методи передачі даних в IRC-мережах.

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

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

- Удосконалено метод передачі даних в IRC-мережах.
- Розроблено вітчизняний продукт передачі даних в IRC-мережах, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі передачі даних в IRC-мережах.

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

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

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

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Боти – це програми, які підключаються до IRC сервера й можуть виконувати спеціальні операції. Крім підтримки статусу оператора, боти можуть виконувати функції дошки оголошень, можуть виводити інформацію про користувача, що заходить на канал, і безліч іншого. Відношення до ботів у різних мережах різне й в основному залежить від загальної політики мережі. Наприклад в IRCNet боти не вітаються, але й немає реєстрації каналів. У мережі EfNet реєстрації каналів також немає, але до ботів ставляться байдуже. У нашій мережі є реєстрація каналів і поки немає серверної програми, що автоматично привласнює господарям каналу статус оператора каналу, для цієї мети дозволено використовувати ботів, якщо вони не заважають нормальній роботі мережі. Втім, підключення бота без особою на те необхідності теж не вітається.

Оператори мережі – це адміністратори, що управляють роботою мережі в цілому. У їхній владі викинути користувача із сервера й заборонити йому вхід на окремий сервер або взагалі на все сервера мережі. Однак операторам заборонено втручатися у внутрішні справи каналів – для цього існують оператори каналів. Фактично оператори можуть втручатися тільки якщо в результаті дій якогось користувача порушується нормальна робота мережі. У нашій мережі оператор мережі також може втручатися в результаті захвата каналу.

1.2 Область застосування

Областю застосування системи є мережі Інтернет. IRC-клієнт – це клієнт IRC, який можна використовувати для спілкування з іншими людьми в мережах IRC по всьому світу. Щоб використовувати IRC-клієнт, вам потрібно підключитися до мережі IRC. Мережа IRC складається з одного або кількох серверів IRC, які підключені один до одного. Підключившись до будь-якого з

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

серверів певної мережі, ви побачите ті самі канали та людей.

Мережі IRC створюють, володіють і керують тисячі незалежних осіб і організацій у всьому світі, і існує багато сотень, якщо не тисяч, таких мереж.

IRC-клієнт поставляється зі списком мереж і серверів IRC за замовчуванням. Якщо ви використовуєте старий список серверів, ви можете оновити свої сервери, завантаживши останній файл servers.ini та зберігши його в папці, де ви встановили IRC-клієнт.

Підключення до мережі IRC

Якщо ви вже встановили IRC-клієнт, ви також можете підключитися до мережі, натиснувши посилання чату, подібні до наведених нижче.

Популярні мережі IRC

Кількість мереж IRC постійно змінюється, оскільки мережі постійно створюються та руйнуються. Деякі мережі IRC існують багато років, а інші зникають через кілька місяців. На даний момент деякі з найбільш популярних мереж IRC:

DALnet	http://www.dal.net	irc://dalnet
EFnet	http://www.efnet.org	irc://efnet
GameSurge	http://www.gamesurge.net	irc://gamesurge
IRCnet	http://www.ircnet.org	irc://ircnet
Libera.Chat	http://www.libera.chat	irc://libera.chat
QuakeNet	http://www.quakenet.org	irc://quakenet
Rizon	http://www.rizon.net	irc://rizon
Snoonet	http://www.snoonet.org	irc://snoonet
SwiftIRC	http://www.swiftirc.net	irc://swiftirc
Undernet	http://www.undernet.org	irc:/ /підмережа

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

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

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

Вебім

Один з найстарших чатів, почав відлік свого існування з 2006 року.



Рисунок 2.1 – Інтерфейс користувача Вебім

Це був один з перших чатів, яким ми користувалися для підтримки клієнтів і який згодом замінили на чат від Кауако. За ті кілька років, що ми не користувалися сервісом він істотно змінився.

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

Особливості:

- Можливість роботи на високонавантажених сайтах (1 млн хітів у добу, 2000 чатів одночасно).
- Кнопка веб-дзвінка в чаті й в окремому віджеті з можливістю прийому дзвінка через програму або SIP.
- Інтеграційний API з CRM -системами, а також інтеграція з Google Analytics.
- Система оцінки конверсії інтернет-трафіку.

Krible

Сервіс активно розвивається з 2009 року. Коротенько зупинимося на конкурентних перевагах.

Особливості:

- Автовідповідач – якщо менеджер відійшов чат привітає відвідувача, а через якийсь час попросить залишити свої контакти для зв'язку.
- Відеозаставка – дозволяє програвати відео із прозорим тлом, для залучення уваги.
- Можливість використання свого сервера для проведення діалогів і зберігання історії.
- Телепортатор – додаток, за допомогою якого менеджери можуть, розмовляючи по телефоні, демонструвати покупцеві товари прямо на його комп'ютері, у браузері. Доступний на тарифах, починаючи від 100 у.о. на місяць.

Cloudim

Є безкоштовний тариф, з підтримкою одного оператора. Платні тарифи починаються від 16 у.о. за оператора на місяць.

Особливості:

- Робота через Jabber.
- Активні запрошення.
- Десктопний клієнт є, зроблений на Adobe Air.

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

GoTalk

Є нативний додаток під Windows і Linux, а для інших операційних систем рекомендують використовувати Jabber клієнти. Наскільки добре працює підтримка Jabber ми не перевіряли, але заявлена підтримка IM+, QIP, PSI

Особливості:

– Наскрізний чат – тобто клієнт завжди впізнається навіть через півроку й переписка із клієнтом доступна за увесь час. Зважаючи на все це реалізовано через cookie.

– Словник термінів – раніше бачив таку опцію тільки в одному чаті. Дозволяє скласти словник термінів і коли в чаті вживається слово, що є в словнику воно підсвічується й при наведенні мишкою на слово можна побачити його значення.

– Оригінал – ця опція дозволяє забрати звідусіль копірайти Gotalk, перемінити доменне ім'я в адресі чату на своє й замінити іконку favicon на свою.

JivoSite

Безкоштовну версію цього чату ми якийсь час тестували в себе на сайті. Варто відзначити просту установку, зручний і приємний інтерфейс.

Коли ми встановили на сайті цей чат, та кількість звернень у чат – істотно зросла, по двох причинах. По-перше через більшу помітність, по-друге завдяки опції активні запрошення. У ній можна налаштувати правила спрацьовування, наприклад, якщо людина перебуває на сторінці з послугою хостингу більше 30 секунд, то можна налаштувати автоматична поява віконця чату з питанням: «Я бачу ви цікавитесь послугою хостингу, можливо у вас є питання?».

Саме ця проста опція дозволяє «чіпляти» значно більше потенційних покупців. До речі кажучи, зараз у всій більшій кількості чатів спочатку з'являтися ця опція.

Особливості:

– Активні запрошення – згадані вже вище. Особливо радує можливість, не тільки вручну запрошувати відвідувача в чат, але й набудувати правила спрацьовування таких запрошень.

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

– Перехід у перерву – дозволяє перемикає оператора в статус перерва якщо комп'ютер не використовувався п хвилин.

– Дзвінки із сайту – от це вже тягне на «киллер-фічу», принаймні така опція є далеко не в усіх. Дозволяє клієнтові зробити дзвінок прямо із чату в службу підтримки. Для дзвінка буде потрібно встановлений флеш-плеєр і гарнітура.

– Клієнт для mac – цю опцію теж можна сміло виносити в переваги, тому що в більшості чатів клієнтів є максимум під windows і linux.



Рисунок 2.2 – Інтерфейс користувача JivoSite

LiveTex

За функціоналом дуже схоже на попередній чат Jivosite. Тут теж у наявності активні запрошення, дзвінки із сайту, клієнт під windows, mac і linux. У загальному все в чаті зроблено приємно й смачно, от тільки ціна не радує. 56 у.о. на місяць за 1 оператора – це самий дорогий чат. Почасти таку ціну виправдує функція дзвінків із сайту, але в конкурентів з аналогічним функціоналом є рішення й дешевше.

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

Особливості:

- Дзвінки й переадресація на SIP.
- Генератор лідів – так отут назвали звичайні оффлайн повідомлення, які можна залишити, коли операторів немає онлайн.
- API – наявність API дозволяє реалізувати деякі нестандартні опції, яких немає за замовчуванням
- Пречат поля – можна додавати довільні поля, які відвідувач повинен заповнити перед початком діалогу, наприклад номер замовлення.
- Утримуючі повідомлення – якщо оператор якийсь час не відповідає, то можна налаштувати показ повідомлень типу «очікуйте будь ласка вільного оператора», або налаштувати автоматичне перемикання на вільного оператора.

Onicon

Особливості:

- Схований режим – дозволяє вручну або по певних умовах приховувати кнопку онлайн чату.
- База знань – дозволяє сформувати список заздалегідь заготовлених відповідей на часті питання.
- Живі форми – дозволяє підключити вже існуючі форми на сайті. Тобто у Вас на сайті є форма зворотного зв'язка, або замовлення в магазині. Ви включаєте «Живі форми» і всі заповнення цих форм починають приходити прямо в Онікон у режимі реального часу.
- Історія діалогів видна для відвідувача при повторному візиті.

R3chat

Клієнта як такого ні, але зате заявлена підтримка великої кількості месенджерів, серед них:

- Клієнти XMPP/Jabber. Google Talk, iChat, eBuddy, Exodus і т.д.
- Клієнти ICQ. ICQ, QIP, AIM, Trillian, Miranda IM і т.д.
- Клієнти YIM. Yahoo! Messenger (YIM), Pidgin (Gaim), Meebo і т.д.

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

Особливості:

– У першу чергу це підтримка великого числа різних месенджерів, що рятує від необхідності в установці окремого клієнта, хоч і обмежує у функціональності.

– Мультимовність – цю опцію варто також відзначити окремо, тому що чат може похвастатися перекладом на 21 мову, серед яких навіть такі екзотичні, як каталанський або галісійський.

RedHelper

Теж свого роду новачок серед чат-сервісів, з'явився він в 2011 році.

Відразу в очі кидається досить оригінальний дизайн сайту, та й сам чат зроблений також дуже приємно.

Особливості:

– Додаток оператора дозволяє спостерігати за клієнтом і підсвітлювати елементи сайту в його браузері.

– API дає можливості повного рестайлінгу чату під дизайн сайту й більшість необхідних функцій інтеграції.

– Спостереження за печаткою – функція дозволяє бачити, що друкує відвідувач ще до того, як він відправив повідомлення.

– Сайт очами відвідувача – у додатку оператора можна спостерігати як переміщається по сайті відвідувач.

SiteHelp

Мабуть ціна й підтримка Jabber і є основним перевагами даного чату. Дизайн досить простий, без особливих изысков, є й функція запрошення відвідувача в чат і автовідповідь і детальна інформація про відвідувача, у загальному функціонал досить стандартний.

Виникає тільки одне питання за рахунок чого система буде монетизуватися й розвиватися. Поки не ясно.

Особливості:

– Чат повністю безкоштовний.

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

SPEXE

Особливості:

- Передзвонимо на мобільний – функція замовлення зворотного дзвінка з перевіркою власника номера по смс.
- Вибір оператора – відвідувач може сам вибрати з яким оператором він хоче поговорити.
- Skype-кнопка виклику оператора з перевіркою статусу зазначених skype-логінів.
- Інтеграція із системою відкликань Userecho.
- Чат може бути додатком для Вконтакті.
- Зручно те що кілька сервісів інтегровані в одній кнопці. Чат, зворотний дзвінок, форма відкликань і побажань і дзвінок скайп.

WebConsult

Функціонал стандартний для чатів. Дизайн досить приємний + можна його налаштувати через візуальний редактор. Оператори можуть працювати тільки через веб-клієнт і Jabber, додатка під Windows немає.

Особливості:

- Pulse – спостереження за менеджерами в реальному часі, за допомогою тимчасової шкали, що рухається, з подіями.
- Розподіл клієнтів між консультантами – можна налаштувати кілька рівнів підтримки. Наприклад: за замовчуванням всі клієнти попадають на рівень один, а якщо на першому рівні всі оператори оффлайн, то вони попадають на другий рівень.

WebConsultant

Мабуть одна з головних особливостей цього чату в тому, що він поширюється не на базі абонентської плати, а доступний за одноразовий платіж, за суму близько 160 у.о.

Наступна перевага полягає в тому, що система встановлюється повністю на ваш сервер, а відповідно до всієї переписки маєте доступ тільки ви.

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

Клієнтови доступно й передача файлу й відправлення діалогу на е-мейл і оцінка оператора.

Дизайн і інтерфейс чату простий і зрозумілий, користуватися ним зручно. Але на жаль для нас головний недолік – це відсутність якого-небудь десктопного додатку.

Особливості:

– Чат установлюється повністю на ваш сервер, а відповідно ви одержуєте повну незалежність від сторонніх сервісів.

– Яких або особливостей, крім можливості купити довічну ліцензію не виявили.

Примітка:

– Особливо виносилися ті функції, які можна назвати конкурентною перевагою, тобто або ті, котрих немає в конкурентів, або ті, які реалізовані особливо вдало.

– Більшість опцій дуже схожі, тому ми не описували їх окремо, а винесли в порівняльну таблицю.

Статистика:

– Варто відзначити, що з 13 протестованих чатів тільки 2 чата (Spexe і GoTalk) відкриваються в новому вікні, інші чати відкриваються в тому же вікні, поверх сторінки. Ми спеціально чи перевірили всі чати, які відкриваються поверх сторінки зберігають діалог при переході на іншу сторінку й повинні відзначити, що із цим завданням усі впоралися.

– Активні запрошення є у всіх чатів, крім r3chat. А от активні запрошення за умовами є не у всіх, тільки в 10.

– Підтримка Jabber є в 9 з 13 чатів.

– Коробкова версія є тільки у двох чатів Вебим і WebConsultant, інші сервіси надають послугу тільки як SaaS.

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

Виводи

Отже з 13 розглянутих ми відразу виключили ряд чатів, що як не підходять нам. Виключили: gotalk і spexe – оскільки чат відкривається в новому вікні, sitehelp – оскільки в нього немає явних переваг, крім безкоштовності. WebConsultant, WebConsult і onicon через відсутність десктопного windows клієнта (хоча варто відзначити, що в WebConsult є підтримка Jabber). p3chat – через відсутність активних запрошень.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

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

Delphi 10.4 Sydney

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

Основні можливості Delphi 10.4.1:

– Істотні розширення для Windows: поліпшення для застосунків на моніторах 4K High DPI, інтеграція з новим WebView2 на базі Chromium,

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

RAD Studio 10.4 Короткий огляд:

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

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

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

Істотне поліпшення Delphi Code Insight

Як найбільше й головне поліпшення інструментів програмування Delphi за багато років, в 10.4 Delphi Code Insight реалізований через Language Server Protocol (LSP). LSP – це технологія генерації результатів для code completion, навігації й інших сервісів в окремому процесі. Це значить, що code completion і Code Insight одержать більш точні результати без блокування IDE. 10.4

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

забезпечує набагато більш високу продуктивність розроблювачів, які працюють із більшими проектами, що містять мільйони рядків коду.

Delphi Custom Managed Records

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

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

Єдине керування пам'яттю

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

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

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

Розширена підтримка бібліотек C++

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

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

Win 64-відладник і збирач для C++

В 10.4 з'явився новий відладник C++ для Windows 64-bit. Відладник заснований на LLDB і показує значне збільшення стабільності при налагодженні

					VKPM-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

64-bit застосунків поряд з новими відладочними можливостями, такими як перегляд і інспекція типів начебто рядків C++ і Delphi, а також колекцій STL, включаючи `std::vector`, `std::map` і інших. Крім того, згенерована для застосунку відладочна інформація має інший внутрішній формат, сприяючи більш стабільному й багатому на можливості процесу налагодження, більш докладним перегляду й інспекції в debug-time.

Підвищення якості й швидкодії інструментів

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

Змінені стилі VCL для High DPI

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

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

Нові High DPI стилі й стилізація окремих VCL компонент

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

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

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

Поліпшена кроссплатформеність

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

Оновлений менеджер пакетів Getit

Менеджер пакетів Getit в IDE був значно вдосконалений.

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

Універсальний інсталятор для установки Online і Offline

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

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

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи передачі даних в IRC-мережах.

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

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

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

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

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

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

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

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

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

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ док.ум.	Підпис	Дата		22

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Незважаючи на такі сучасні альтернативи, як Slack, стародавній *IRC* все ще надзвичайно популярний як інтерактивна платформа онлайн-чату.

Це може бути тому, що існують IRC-клієнти майже для кожної операційної системи та пристрою, від Commodore Amiga до вашого смартфона, а технологія, що лежить в основі IRC, надихає простотою – це просто необроблений текст і кілька керуючих символів, які переміщуються по мережі.

Онлайн-сервери, такі як сервери Libera Chat , чудово підходять як для публічних, так і для приватних каналів. Але так само легко запустити власний приватний IRC-сервер, що дає вам повний контроль над своїми даними, журналами та параметрами конфігурації, уникаючи всіх ризиків і розчарувань, пов'язаних зі спамерами та ботами IRC.

У цьому розділі ми розглянемо встановлення IRC-сервера InspIRCd на Ubuntu, від встановлення його залежностей і створення останньої версії з GitHub до налаштування та виконання.

Для першого кроку ми встановимо залежності, необхідні для створення та запуску InspIRCd, починаючи з *git* :

```
sudo apt-get install git
```

Далі йде *Perl* , щоб ми могли запустити сценарій конфігурації, який входить до InspIRCd:

```
sudo apt-get install perl
```

```
sudo apt-get install g++
```

І, нарешті, нам потрібно переконатися, що *make* встановлено:

```
sudo apt-get install make
```

Останню версію InspIRCd можна завантажити з:

<https://github.com/inspircd/inspircd/releases/latest>

									Арк.
									23
Вим.	Арк.	№ докум.	Підпис	Дата	ВКРМ-122.23.0070.00.00.ПЗ				

Оскільки ми збираємося створювати останню версію з вихідного коду, нам потрібно отримати архів *tar.gz* за допомогою вашого браузера або з командного рядка. Щоб взяти версію 2.0.25 як приклад, ви можете використати наступну команду для завантаження архіву:

```
wget
```

```
https://github.com/inspircd/inspircd/archive/v2.0.25.tar.gz
```

Використовуйте *tar*, щоб розпакувати завантаження:

```
tar xvf ./v2.0.25.tar.gz
```

Завантаживши та розпакувавши вихідний код, тепер ми можемо налаштувати спосіб створення InspIRCd.

Спочатку увійдіть в каталог встановлення:

```
cd inspircd-2.0.25
```

Номер версії вище має відповідати завантаженій версії.

Щоб почати налаштування встановлення, введіть наступне:

```
perl ./configure
```

Зараз вам задають низку запитань. Якщо не впевнені, натисніть кнопку *повернення*, щоб відповісти зі значеннями за замовчуванням.

Останнє запитання питатиме вас про те, чи бажаєте ви перевірити наявність оновлень для сторонніх модулів, і ви повинні відповісти утак.

Кінцевий результат має бути приблизно таким:

```
Ok, 144 modules.  
Writing inspircd_config.h  
Writing GNUmakefile ...  
Writing BSDmakefile ...  
Writing inspircd ...  
Writing cache file for future ./configures ...
```

To build your server with these settings, please run 'make' now.

```
*** Remember to edit your configuration files!!! ***
```

Тепер ми можемо продовжити етап *побудови*.

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

Створіть сервер

Тепер сервер можна створити, виконавши його makeв каталозі інсталяції. Цей процес займе приблизно 10 хвилин, залежно від швидкості вашої системи, тому сміливо відходьте від комп'ютера.

Закінчивши, введіть, make installщоб перемістити виконувані файли до налаштованих раніше місць. Огляд цього процесу виводиться після завершення:

```
*****
```

```
*          BUILDING INSPIRCD          *
```

```
*                                     *
```

```
*   This will take a *long* time.   *
```

```
*   Why not read our wiki at       *
```

```
*   http://wiki.inspircd.org         *
```

```
*   while you wait for make to run? *
```

```
*****
```

```
*****
```

```
*          INSTALL COMPLETE!        *
```

```
*****
```

Paths:

Base install: /home/javier/build/inspircd-2.0.25/run

Configuration: /home/javier/build/inspircd-2.0.25/run/conf

Binaries: /home/javier/build/inspircd-2.0.25/run/bin

Modules: /home/javier/build/inspircd-2.0.25/run/modules

Data: /home/javier/build/inspircd-2.0.25/run/data

To start the ircd, run: /home/javier/build/inspircd-2.0.25/run/inspircd start

Remember to create your config file: /home/javier/build/inspircd-2.0.25/run/conf/inspircd.conf

Examples are available at: /home/javier/build/inspircd-2.0.25/run/conf/examples/

Конфігурація сервера

У каталозі збірки створіть текстовий файл із назвою run/config/inspircd.confта вставте наступне:

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

```

<config format="xml">
<define name="bindip" value="1.2.2.3">
<define name="localips" value="&bindip;/24">

##### SERVER CONFIGURATION #####

<server
    name="SERVER_HOSTNAME/FQDN"
    description="SERVER_DESCRIPTION"
    id="SERVER_SID"
    network="NETWORK_NAME">

##### ADMIN INFO #####

<admin
    name="ADMIN_NAME"
    nick="ADMIN_NICK"
    email="ADMIN_EMAIL">

##### PORT CONFIGURATION #####

<bind
    address="SERVER_IP"
    port="SERVER_PORT"
    type="SERVER_TYPE">

```

Змініть наступні значення в тексті вище, щоб відобразити вашу власну конфігурацію:

- *SERVER_HOSTNAME/FQDN* : Ім'я хосту для сервера
- *SERVER_DESCRIPTION* : опис вашого сервера
- *SERVER_SID* : унікальна послідовність із 3 символів, перший з яких є числом (переконайтеся, що введено великі літери)
- *NETWORK_NAME* : назва вашої мережі IRC

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

- *ADNIN_NAME* : ім'я адміністратора IRC
- *ADMIN_NICK* : псевдонім адміністратора IRC
- *ADMIN_EMAIL* : електронна адреса адміністратора IRC
- *SERVER_IP* : загальнодоступна IP-адреса для сервера
- *SERVER_PORT* : порт сервера (зазвичай 6697)
- *SERVER_TYPE* : тип клієнтів або серверів (тут клієнти мають бути добре)

Тепер файл конфігурації має виглядати приблизно так:

```
<config format="xml">
<define name="bindip" value="1.2.2.3">
<define name="localips" value="&bindip;/24">

##### SERVER CONFIGURATION #####

<server
    name="tutorials.ubuntu.com"
    description="Welcome to Ubuntu Tutorials"
    id="97K"
    network="tutorials.ubuntu.com">

##### ADMIN INFO #####

<admin
    name="tutorial ubuntu"
    nick="tutorial"
    email="tutorials@ubuntu.com">

##### PORT CONFIGURATION #####

<bind
    address="23.54.785.654"
    port="6697"
```

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

```
type="clients">
```

Обов'язково збережіть зміни!

Настав час уперше запустити InspIRCd!

У вікні терміналу введіть:

```
./inspircd start  
./inspircd status
```

У разі успіху ви побачите такий результат:

```
InspIRCd is running (PID: 13301)
```

Щиро вітаю! Ваш сервер тепер онлайн!

Будь-який клієнт IRC, який має доступ до вашого сервера, тепер зможе підключитися до вашого сервера IRC.

Тепер ви знаєте, як:

- Візьміть вихідний код з GitHub.
- Підготуйте середовище для створення InspIRCd.
- Налаштуйте InspIRCd за допомогою .confфайлу.
- Запустіть службу.

Подальші дії:

- Змініть зовнішнє підключення до Інтернету, щоб перенаправити порт 6697.
- Переконайтеся, що ваша мережа має статичну IP-адресу.
- Отримайте URL-адресу для легкого надсилання трафіку на ваш сервер (необов'язково).

Зміни на вашому IRC-сервері потрібно внести у файл конфігурації, а список підтримуваних команд та іншу корисну інформацію можна знайти на Wiki InspIRCd .

Допомога:

- Двічі перевірте, чи доступний порт.
- Переконайтеся, що .configфайл правильний.
- Переконайтеся, що ви правильно ввели команди.
- Спробуйте скористатися sudo (якщо ви ще цього не зробили).

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

- Поставте запитання на Ask Ubuntu.

3.2 Розробка структурної схеми

Члени мережі IRC діють як плацдарм для передачі технологій та інновацій з метою підвищення конкурентоспроможності європейських малих і середніх підприємств.

IRC мають шість рекомендованих завдань:

- Транснаціональна передача технологій.
- Транснаціональне поширення та використання результатів досліджень Співтовариства.
- Стимулювати здатність фірм до впровадження нових технологій.
- Сприяти спільним транснаціональним інноваційним ініціативам.
- Надати інформацію про інноваційні заходи FP5.
- Вказівки на інші служби.

У 1995 році Європейська комісія створила мережу IRC. З квітня 2000 року він складається з 68 інноваційних релейних центрів (IRC) по всій Європі, включаючи ЄС, Болгарію, Чеську Республіку, Кіпр, Естонію, Угорщину, Ісландію, Ізраїль, Латвію, Литву, Мальту, Норвегію, Польщу, Румунію, Словаччину, Словенію та Швейцарію. Ці центри були створені для того, щоб сприяти передачі інноваційних технологій європейським компаніям або дослідницьким відділам і з них. Ці центри були створені для того, щоб сприяти передачі інноваційних технологій європейським компаніям або дослідницьким відділам і з них. Будучи рушієм інновацій, мережа IRC стала провідною європейською мережею для сприяння технологічному партнерству та передачі в основному між малими та середніми компаніями (МСП). IRC є постачальниками послуг підтримки інновацій, які в основному обслуговують громадські організації, такі як університетські технологічні центри, торгові палати, агенції регіонального розвитку або національні інноваційні агенції. Більшість IRC створено як консорціуми. У кожному центрі працюють співробітники, які мають

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

глибокі знання технологічного та економічного профілю компаній і регіонів, які вони обслуговують.

Пересилання повідомлень в IRC-мережі

IRC надає можливість як групового, так і приватного спілкування. Для групового спілкування існує кілька можливостей. Користувач може відправити повідомлення списку користувачів, при цьому серверу відправляється список, сервер виділяє з нього окремих користувачів і відправляє копію повідомлення кожному з них. Більше ефективним є використання каналів. У цьому випадку повідомлення відправляється безпосередньо серверу, а сервер відправляє його всім користувачам у каналі. Як при груповому, так і при приватному спілкуванні повідомлення відправляються клієнтам по найкоротшому шляху й видимі тільки відправникові, одержувачеві й вхідним в найкоротший шлях серверам.

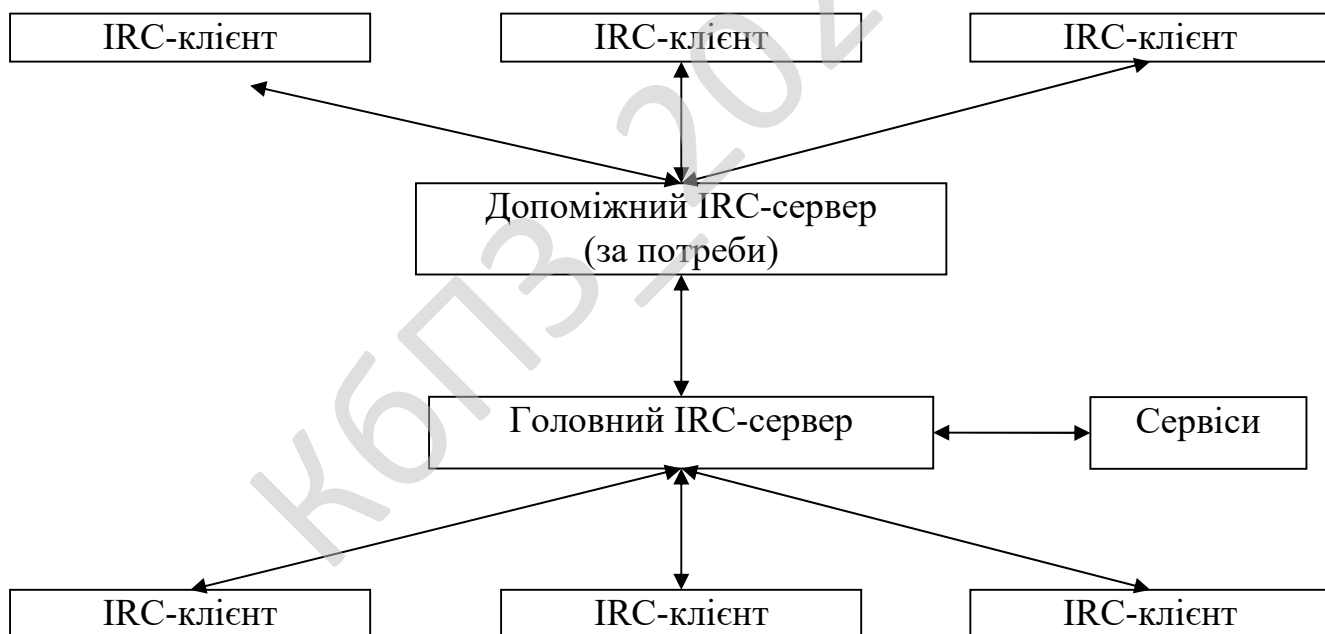


Рисунок 3.1 – Структурна схема системи

Крім того, можливе відправлення широкомовного повідомлення. Повідомлення клієнтів, що стосуються зміни стану мережі (наприклад, режиму каналу або статусу користувача) повинні відправлятися всім серверам, що

входять у мережу. Всі повідомлення, що виходять від сервера, також повинні бути відправлені всім іншим серверам.

Користувальницькі режими

- i – невидимий користувач;
- s – одержувати повідомлення сервера;
- w – одержувати широкомовні повідомлення;
- – оператор сервера.

Стан каналів

Оператор каналу може міняти його стан за допомогою команди, додаючи/видаляючи різні прапори як каналу, так і окремим користувачам у каналі.

Користувальницькі прапори

- user – позначає творця каналу (вручну змінити неможливо);
- user – позначає оператора каналу;
- h user – позначає напівоператора каналу;
- v user – дає користувачеві право відправляти повідомлення в модеруємий канал.

Прапори каналу

- a – анонімний канал (імена всіх користувачів ховаються);
- i – доступ у канал тільки за запрошенням;
- m – модеруємий канал (тільки користувачі із прапорами +o, +h, або +v можуть посилати в цей канал повідомлення);
- n – тільки користувачі, що перебувають у каналі, можуть посилати в нього повідомлення;
- q – «тихий» канал (обмеження на відправлення серверних повідомлень);
- p/s – приватний/секретний канал (не відображається в загальному списку каналів, у нього можна потрапити тільки знаючи ім'я каналу);
- r – якщо в каналі протягом певного часу відсутні всі оператори, сервер сам призначає операторів із числа користувачів у каналі;

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

- t – тема каналу може бути змінена тільки оператором каналу;
- k – пароль каналу;
- l – ліміт користувачів у каналі;
- b mask – заборона на доступ до каналу користувачів, що збігаються з маскою mask;
- e mask – дозволяє користувачам, що попадають під маску mask, заходити на канал поза залежністю від прапора b;
- i mask – дозволяє користувачам, що попадають під маску mask, заходити на канал поза залежністю від режиму i.

Якщо ви вже знаєте, як технічно виглядає структура IRC мережі, то пропустите теоретичний розділ.

Структура IRC-мережі

Представимо структурну схему IRC мережі. Тут об'єктами є додатки (не комп'ютери). Технічно все це може розташовуватися й на одному комп'ютері. Напрямок стрілок означає порядок з'єднання (хто до кого підключається).

Тут ви бачите 2 сервери, одні сервіси, і клієнтів, підключених до сервера. Тепер поясню, навіщо всього так багато, і як все це працює. Перше, про що обмовлюся, це кількість серверів. У вашій мережі повинен бути як мінімум один сервер. У більшості випадків його одного буде досить. Більша кількість серверів потрібна, якщо ви, наприклад, хочете об'єднати чат у своїй локальній мережі із чатом у сусідній мережі. Далі буде обговорюватися мережа, що складається з одного сервера (але все сказане справедливо й для багатосерверної мережі).

Сервер (IRC демон, IRCD). Для чого потрібний сервер, думаю, зрозуміло. До нього підключаються клієнти. Сервер типово працює на 6667 порту, тому для його запуску не потрібні (а також не рекомендуються) права суперкористувача (root). Серверне ПЗ кроссплатформене: ви можете його використовувати як на *nix-подібних операційних системах, так і на Windows, також воно не вибагливо до заліза сервера (підійде й старий домашній комп'ютер). Налаштування серверного ПЗ буде розглянута в розділі налаштування.

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

Ваш сервер повинен бути доступний 24 години на добу, 7 днів у тиждень.
У момент його відсутності люди не зможуть спілкуватися через IRC.

3.3 Розробка функціональної схеми

Для того, щоб описати функціональну схему системи, наведемо перелік команд, які використовуються для передачі даних в IRC-мережах.

Реєстрація й підключення

Рекомендується використовувати команди в наступному порядку: PASS -> SERVICE або PASS -> NICK -> USER.

PASS

Параметри: <password>.

Установлює пароль з'єднання.

Можливі відгуки сервера:

- ERR_NEEDMOREPARAMS.
- ERR_ALREADYREGISTERED.

NICK

Параметри: <nickname>.

Зміна ніку.

Можливі відгуки сервера:

- ERR_NONICKNAMEGIVEN.
- ERR_ERRONEUSNICKNAME.
- ERR_NICKNAMEINUSE.
- ERR_NICKCOLLISION.
- ERR_UNAVAILRESOURCE.
- ERR_RESTRICTED.

SERVICE

Параметри: <nickname> <reserved> <distribution> <type> <reserved> <info>.

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

Реєстрація нового клієнта-сервісу.

Можливі відгуки сервера:

- ERR_ALREADYREGISTRED.
- ERR_NEEDMOREPARAMS.
- ERR_ERRONEUSNICKNAME.
- RPL_YOURESERVICE.
- RPL_YOURLHOST.
- RPL_MYINFO.

USER

Параметри: <user> <mode> <unused> <realname>.

Використовується на початку з'єднання для визначення нового користувача.

Можливі відгуки сервера:

- ERR_NEEDMOREPARAMS.
- ERR_ALREADYREGISTRED.

OPER

Параметри: <name> <password>.

Одержання статусу оператора.

Можливі відгуки сервера:

- ERR_NEEDMOREPARAMS.
- RPL_YOUREOPER.
- ERR_NOOPERHOST.
- ERR_PASSWDMISMATCH.

MODE

Параметри: <nickname> *(<"+" / «->> <flag>).

Зміна стану користувача – додавання/видалення прапорів; команда доступна тільки операторам.

Можливі відгуки сервера:

- ERR_NEEDMOREPARAMS.

					БКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

- ERR_USERSDONTMATCH.
- ERR_UMODEUNKNOWNFLAG.
- RPL_UMODEIS.

QUIT

Параметри: [<Quit Message>].

Вихід користувача з мережі.

Можливі відгуки сервера: відсутні.

SQUIT

Параметри: <server> <comment>.

Вихід сервера з мережі; команда доступна тільки операторові сервера або серверу при виникненні помилки.

Можливі відгуки сервера:

- ERR_NOPRIVILEGES.
- ERR_NOSUCHSERVER.
- ERR_NEEDMOREPARAMS.

Робота з каналами

JOIN

Параметри: <channel> *(" " <channel>) [<key> *(" " <key>)].

Вхід у канал.

Можливі відгуки сервера:

- ERR_NEEDMOREPARAMS.
- ERR_BANNEDFROMCHAN.
- ERR_INVITEONLYCHAN.
- ERR_BADCHANNELKEY.
- ERR_CHANNELISFULL.
- ERR_BADCHANMASK.
- ERR_NOSUCHCHANNEL.
- ERR_TOOMANYCHANNELS.
- ERR_TOOMANYTARGETS.

– ERR_UNAVAILRESOURCE.

– RPL_TOPIC.

PART

Параметри: <channel> *(" , «<channel>) [<Part Message>].

Вихід з каналу.

Можливі відгуки сервера:

– ERR_NEEDMOREPARAMS.

– ERR_NOSUCHCHANNEL.

– ERR_NOTONCHANNEL.

MODE

Параметри: <channel> *((,,-,," / ,,+“) *<modes> *<modeparams>).

Зміна режиму каналу; у команді допускається не більше трьох режимів з параметрами.

Можливі відгуки сервера:

– ERR_NEEDMOREPARAMS.

– ERR_KEYSET.

– ERR_NOCHANMODES.

– ERR_CHANOPRIVSNEEDED.

– ERR_USERNOTINCHANNEL.

– ERR_UNKNOWNMODE.

– RPL_CHANNELMODEIS.

– RPL_BANLIST.

– RPL_ENDOFBANLIST.

– RPL_EXCEPTLIST.

– RPL_ENDOFEXCEPTLIST.

– RPL_INVITELIST.

– RPL_ENDOFINVITELIST.

– RPL_UNIQOPIS.

TOPIC

Параметри: <channel> [»: " <topic>].

Зміна або перегляд теми каналу.

Можливі відгуки сервера:

- ERR_NEEDMOREPARAMS.
- ERR_NOTONCHANNEL.
- RPL_NOTOPIC.
- RPL_TOPIC.
- ERR_CHANOPRIVSNEEDED.
- ERR_NOCHANMODES.

NAMES

Параметри: [<channel> *(" " <channel>) [<target>]].

Переглянути список імен користувачів, підключених до сервера target і присутніх у каналі channel.

Можливі відгуки сервера:

- ERR_TOOMANYMATCHES.
- ERR_NOSUCHSERVER.
- RPL_NAMREPLY.
- RPL_ENDOFNAMES.

LIST

Параметри: [<channel> *(" " <channel>) [<target>]].

Перегляд списку каналів/статусу зазначених каналів.

Можливі відгуки сервера:

- ERR_TOOMANYMATCHES.
- ERR_NOSUCHSERVER.
- RPL_LIST.
- RPL_LISTEND.

INVITE

Параметри: <nickname> <channel>.

Запросити користувача в канал.

Можливі відгуки сервера:

- ERR_NEEDMOREPARAMS.
- ERR_NOSUCHNICK.
- ERR_NOTONCHANNEL.
- ERR_USERONCHANNEL.
- ERR_CHANOPRIVSNEEDED.
- RPL_INVITING.
- RPL_AWAY.

KICK

Параметри: <channel> *(", " <channel>) <user> *(", " <user>) [<comment>].

Видалення користувача з каналу.

Можливі відгуки сервера:

- ERR_NEEDMOREPARAMS.
- ERR_NOSUCHCHANNEL.
- ERR_BADCHANMASK.
- ERR_CHANOPRIVSNEEDED.
- ERR_USERNOTINCHANNEL.
- ERR_NOTONCHANNEL.

Відправлення повідомлень

PRIVMSG

Параметри: <msgtarget> <text to be sent>.

Відправлення особистого (приватного) повідомлення.

Можливі відгуки сервера:

- ERR_NORECIPIENT.
- ERR_NOTEXTTOSEND.
- ERR_CANNOTSENDTOCHAN.
- ERR_NOTOPLEVEL.
- ERR_WILDTOplevel.

- ERR_TOOMANYTARGETS.
- ERR_NOSUCHNICK.
- RPL_AWAY.

NOTICE

Параметри: <msgtarget> <text>.

Команда використовується аналогічно PRIVMSG, але без одержання автоматичної відповіді.

Робота із серверами

MOTD

Параметри: [<target>].

Установка так званого «повідомлення дня».

Можливі відгуки сервера:

- RPL_MOTDSTART.
- RPL_MOTD.
- RPL_ENDOFMOTD.
- ERR_NOMOTD.

LUSERS

Параметри: [<mask> [<target>]].

Відомості про користувачів, підключених до мережі/підмережі, що відповідає зазначеній масці/серверу.

Можливі відгуки сервера:

- RPL_USERCLIENT.
- RPL_USEROP.
- RPL_USERUNKOWN.
- RPL_USERCHANNELS.
- RPL_USERME.
- ERR_NOSUCHSERVER.

VERSION

Параметри: [<target>].

Версія програми-сервера.

Можливі відгуки сервера:

- ERR_NOSUCHSERVER.
- RPL_VERSION.

STATS

Параметри: [<query> [<target>]].

Статистика по серверу.

Можливі відгуки сервера:

- ERR_NOSUCHSERVER.
- RPL_STATSLINKINFO.
- RPL_STATSUPTIME.
- RPL_STATSCOMMANDS.
- RPL_STATSOLINE.
- RPL_ENDOFSTATS.

LINKS

Параметри: [[<remote server>] <server mask>].

Імена серверів, підключених до зазначеного/поточний.

Можливі відгуки сервера:

- ERR_NOSUCHSERVER.
- RPL_LINKS.
- RPL_ENDOFLINKS.

TIME

Параметри: [<target>].

Локальний час сервера.

Можливі відгуки сервера:

- ERR_NOSUCHSERVER.
- RPL_TIME.

CONNECT

Параметри: <target server> <port> [<remote server>].

Спроба підключити поточний сервер до зазначеного через зазначений порт.

Можливі відгуки сервера:

- ERR_NOSUCHSERVER.
- ERR_NOPRIVILEGES.
- ERR_NEEDMOREPARAMS.

TRACE

Параметри: [<target>].

Шлях до зазначеного сервера.

Можливі відгуки:

- ERR_NOSUCHSERVER.
- RPL_TRACELINK.
- RPL_TRACECONNECTING.
- RPL_TRACEHANDSHAKE.
- RPL_TRACEUNKNOWN.
- RPL_TRACEOPERATOR.
- RPL_TRACEUSER.
- RPL_TRACESERVER.
- RPL_TRACESERVICE.
- RPL_TRACENEWTYPE.
- RPL_TRACECLASS.
- RPL_TRACELOG.
- RPL_TRACEEND.

ADMIN

Параметри: [<target>].

Інформація про адміністратора зазначеного/поточного сервера.

Можливі відгуки сервера:

- ERR_NOSUCHSERVER.
- RPL_ADMINME.

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

- RPL_ADMINLOC1.
- RPL_ADMINLOC2.
- RPL_ADMINEMAIL.

INFO

Параметри: [<target>].

Інформація про сервер.

Можливі відгуки сервера:

- ERR_NOSUCHSERVER.
- RPL_INFO.
- RPL_ENDOFINFO.

Робота із клієнтами-сервісами

SERVLIST

Параметри: [<mask> [<type>]].

Список доступних сервісів.

Можливі відгуки сервера:

- RPL_SERVLIST.
- RPL_SERVLISTEND.

SQUERY

Параметри: <servicename> <text>.

Аналогічно PRIVMSG.

Робота з даними користувачів

WHO

Параметри: [<mask> [«о»]].

Даних користувачів, ніки яких відповідають зазначеній масці.

Можливі відгуки сервера:

- ERR_NOSUCHSERVER.
- RPL_WHOREPLY.
- RPL_ENDOFWHO.

WHOIS

Параметри: [<target>] <mask> *(" " <mask>).

Дані про певного користувача.

Можливі відгуки сервера:

- ERR_NOSUCHSERVER.
- ERR_NONICKNAMEGIVEN.
- RPL_WHOISUSER.
- RPL_WHOISCHANNELS.
- RPL_WHOISCHANNELS.
- RPL_WHOISSERVER.
- RPL_AWAY.
- RPL_WHOISOPERATOR.
- RPL_WHOISIDLE.
- ERR_NOSUCHNICK.
- RPL_ENDOFWHOIS.

WHOWAS

Параметри: <nickname> *(" " <nickname>) [<count> [<target>]].

Дані користувача, відсутнього в мережі, тобто вже не існуючого.

Можливі відгуки сервера:

- ERR_NONICKNAMEGIVEN.
- ERR_WASNOSUCHNICK.
- RPL_WHOWASUSER.
- RPL_WHOISSERVER.
- RPL_ENDOFWHOWAS.

Інші команди

KILL

Параметри: <nickname> <comment>.

Розрив з'єднання із зазначеним користувачем з боку сервера.

Можливі відгуки сервера:

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

- ERR_NOPRIVILEGES.
- ERR_NEEDMOREPARAMS.
- ERR_NOSUCHNICK.
- ERR_CANTKILLSERVER.

PING

Параметри: <server1> [<server2>].

Перевірка активності клієнта/сервера на іншій стороні з'єднання.

Можливі відгуки:

- ERR_NOORIGIN.
- ERR_NOSUCHSERVER.

PONG

Параметри: <server> [<server2>].

Відповідь на повідомлення PING від сервера server серверу server2.

Можливі відгуки:

- ERR_NOORIGIN.
- ERR_NOSUCHSERVER.

ERROR

Параметри: <error message>.

Використовується сервером для повідомлення про помилку.

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

Функціональні можливості розробленої системи:

1. Головне вікно інтерфейсу користувача програми.
2. Підтримка модульних розширень завантажуваних і вивантажуваних за потребою.
3. Необмежена можливість розширення за рахунок універсального API.
4. Підтримка одночасного з'єднання з різними IRC-серверами з різною конфігурацією профілю.
5. Упакування логів у фоновому режимі.

6. DCC CHAT (чат) і SEND (з підтримкою дозавантаження).
7. Додаткові DCC: VOICE, RSEND, GET, RECV.
8. DCC CHAT (чат) через SSL.
9. Фонове лімітування швидкості DCC з'єднання.
10. Інтегрований движок скриптингу.
11. Можливість використання об'єктно-орієнтованого програмування.
12. Багатомовний інтерфейс.
13. Підтримка скинів, створення скинів користувачем.
14. Індикація активності IRC.
15. Повна сумісність ядра з Unicode UTF-8.
16. Робота через проксі сервери (SOCKSv4, SOCKSv5, HTTP).
17. Робота з IPv6, включаючи DCC.
18. Використання SSL, включаючи IPv6 і DCC.
19. Шифрування (можна використовувати на каналах, у приваті або в

DCC):

- Micrption.
- Rijndael256Base64.
- Rijndael192Base64.
- Rijndael128Base64.
- Rijndael256Hex.
- Rijndael192Hex.
- Rijndael128Hex.
- Lamerizer.

Розглянемо деякі з понять які визначені на функціональній схемі.

HTTP (англ. HyperText Transfer Protocol – «протокол передачі гіпертексту») – протокол прикладного рівня передачі даних (споконвічно – у вигляді гіпертекстових документів). Основою HTTP є технологія «клієнт-сервер», тобто передбачається існування споживачів (клієнтів), які ініціюють з'єднання й надсилають запит, і постачальників (серверів), які очікують з'єднання для одержання запиту, роблять необхідні дії й повертають назад повідомлення з

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

результатом. HTTP у цей час повсюдно використовується у Всесвітній павутині для одержання інформації з веб-сайтів. В 2012 році в Північній Америці частка HTTP-трафіку перевищила частку P2P-мереж і склала 46 %, з яких майже половина – це передача потокового відео й звуку. HTTP використовується також у якості «транспорту» для інших протоколів прикладного рівня, таких як SOAP, XML-RPC, WebDAV. Основним об'єктом маніпуляції в HTTP є ресурс, на який вказує URI (англ. Uniform Resource Identifier) у запиті клієнта. Звичайно такими ресурсами є файли, що зберігаються на сервері, але ними можуть бути логічні об'єкти або щось абстрактне. Особливістю протоколу HTTP є можливість вказати в запиті й відповіді спосіб подання того самого ресурсу по різних параметрах: формату, кодуванню, мові й т.д. (Зокрема для цього використовується HTTP-заголовок.) Саме завдяки можливості вказівки способу кодування повідомлення клієнт і сервер можуть обмінюватися двійковими даними, хоча даний протокол є текстовим. HTTP – протокол прикладного рівня, аналогічними йому є FTP і SMTP. Обмін повідомленнями йде за звичайною схемою «запит-відповідь». Для ідентифікації ресурсів HTTP використовує глобальні URI. На відміну від багатьох інших протоколів, HTTP не зберігає свого стану. Це означає відсутність збереження проміжного стану між парами «запит-відповідь». Компоненти, що використовують HTTP, можуть самостійно здійснювати збереження інформації про стан, пов'язаної з останніми запитами й відповідями (наприклад, «кукі» на стороні клієнта, «сесії» на стороні сервера). Браузер, що посилає запити, може відслідковувати затримки відповідей. Сервер може зберігати IP-Адреси й заголовки запитів останніх клієнтів. Однак сам протокол не обізнаний про попередні запити й відповіді, у ньому не передбачена внутрішня підтримка стану, до нього не пред'являються такі вимоги.

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

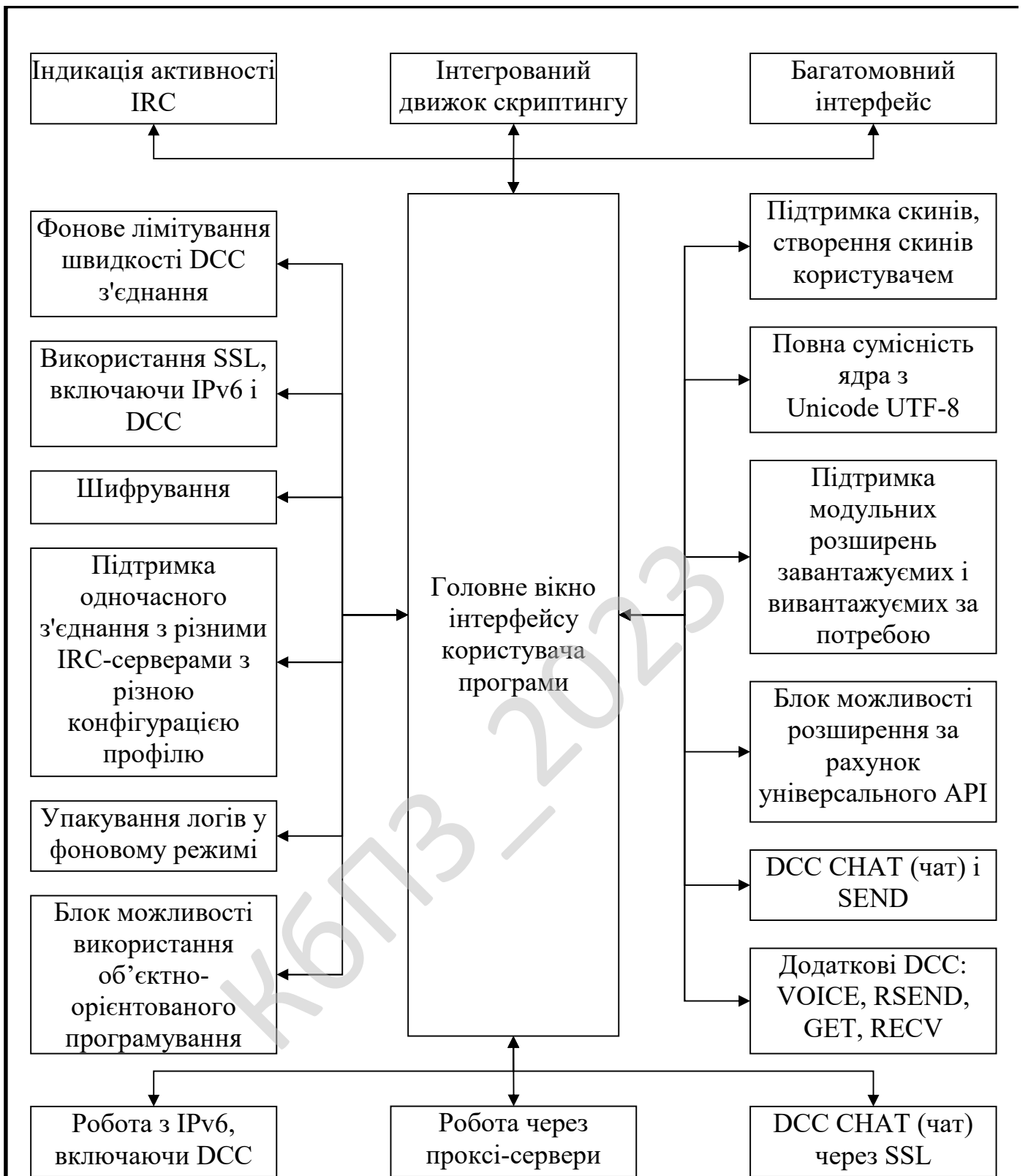


Рисунок 3.2 – Функціональна схема системи

IPv6 (англ. Internet Protocol version 6) – нова версія протоколу IP, покликана вирішити проблеми, з якими зіткнулася попередня версія (IPv4) при її використанні в інтернеті, за рахунок використання довжини адреси 128 біт замість 32. У цей час протокол IPv6 уже використовується в декількох тисячах мереж по усьому світі, але поки ще не одержав настільки широкого поширення в Інтернеті, як IPv4. В Україні використовується майже винятково в тестовому режимі деякими операторами зв'язку, а також реєстраторами доменів для роботи DNS-серверів. Протокол був розроблений IETF. Після того, як адресний простір в IPv4 закінчиться, два стеки протоколів – IPv6 і IPv4 – будуть використовуватися паралельно (англ. dual stack), з поступовим збільшенням частки трафіку IPv6 у порівнянні з IPv4. Така ситуація стане можливою через наявність величезної кількості пристроїв, у тому числі застарілих, не підтримуючих IPv6 і потребуючих спеціального перетворення для роботи із пристроями, що використовують тільки IPv6.

DCC (Direct Client-to-Client) – одне з розширень IRC, що дозволяє робити клієнтам обмін інформацією не через сервер, а прямо, установлюючи TCP-з'єднання між двома клієнтами. Ініціатор з'єднання визначається під протоколом DCC.

SSL (англ. Secure Sockets Layer – рівень захищених сокетів) – криптографічний протокол, що забезпечує встановлення безпечного з'єднання між клієнтом і сервером. SSL споконвічно розроблений компанією Netscape Communications. Згодом на підставі протоколу SSL 3.0 був розроблений і прийнятий стандарт RFC, що одержав ім'я TLS. Протокол забезпечує конфіденційність обміну даними між клієнтом і сервером, що використовують TCP/IP, причому для шифрування використовується асиметричний алгоритм із відкритим ключем. При шифруванні з відкритим ключем використовуються два ключі, відкритий і секретний, причому кожної з них може використовуватися для шифрування повідомлення. Якщо для шифрування повідомлення був використаний відкритий ключ, то для розшифровки повинен використовуватися секретний, і навпаки. У такій ситуації можливі два способи використання ключів.

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

По-перше, сторона, що зберігає в таємниці секретний ключ і яка оприлюднила відкритий, може приймати від протилежної сторони повідомлення, зашифровані відкритим ключем, які не може прочитати ніхто, крім неї (адже для розшифровки потрібен секретний ключ, відомий тільки їй). По-друге, за допомогою закритого ключа сторона-власник закритого ключа може створювати зашифровані повідомлення, які може прочитати хто завгодно (адже для розшифровки потрібний відкритий ключ, доступний усім), але при цьому що прочитав може бути впевнений, що це повідомлення було створено стороною-власником секретного ключа.

Проксі-сервер (від англ. проху – «представник, уповноважений») – служба (комплекс програм) у комп'ютерних мережах, що дозволяє клієнтам виконувати непрямі запити до інших мережних служб. Спочатку клієнт підключається до проксі-сервера й запитує який-небудь ресурс (наприклад, e-mail), розташований на іншому сервері. Потім проксі-сервер або підключається до зазначеного сервера й одержує ресурс у нього, або повертає ресурс із власного кешу (у випадках, якщо проксі має свій кеш). У деяких випадках запит клієнта або відповідь сервера може бути змінений проксі-сервером у певних цілях. Також проксі-сервер дозволяє захищати клієнтський комп'ютер від деяких мережних атак і допомагає зберігати анонімність клієнта.

SOCKS – мережний протокол, що дозволяє клієнт-серверним додаткам прозоро використовувати сервіси за міжмережними екранами (фаєрволами). SOCKS – це скорочення від «SOCKet Secure».

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

3.4 Розробка діаграми процесів

Діаграма процесів розробленої системи зображена на рисунку 3.3

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

- Список користувачів каналу.
- Вибір користувача.
- Виведення вікна приватного чату.
- Архів повідомлень.
- Написання та відправка приватного повідомлення.

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

КБПЗ_2023

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
<i>Вим.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		51

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

Первинною стадією без якої не відбувається розробка програмного забезпечення це звичайно розробка блок-схем.

На рисунку 4.1 зображена основна блок-схема програми, на рисунку 4.2 зображено роботу підпрограми.

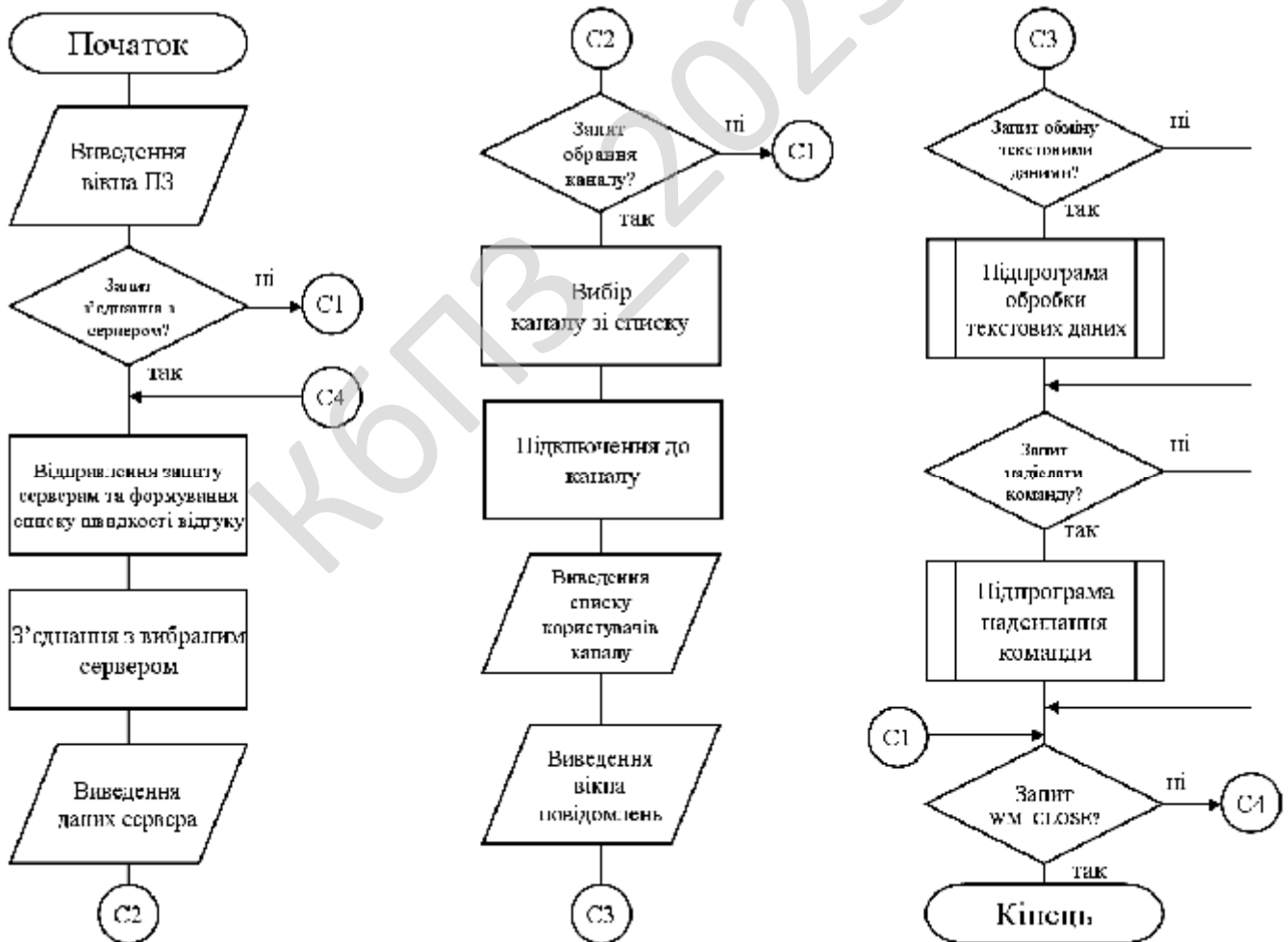


Рисунок 4.1 – Блок схема основної програми

З якої видно що робота основної програми складається з початкових етапів ініціалізації ПЗ, перевірки наявності ресурсів системи, блоку початку основного циклу з чеканням запиту від користувача в якому відбувається виклик підпрограми та останньої стадії – перевірка поточного стану з завершенням роботи розробленого ПЗ. При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання. При складанні блок-схем програмного забезпечення і напрацювання алгоритмів я зіткнувся з масою проблем, які вимагали напрацювання процедур і функцій над основною проблематикою. Для чого були створені додаткові класи, типи даних і константи, що забезпечило вирішення проблем.

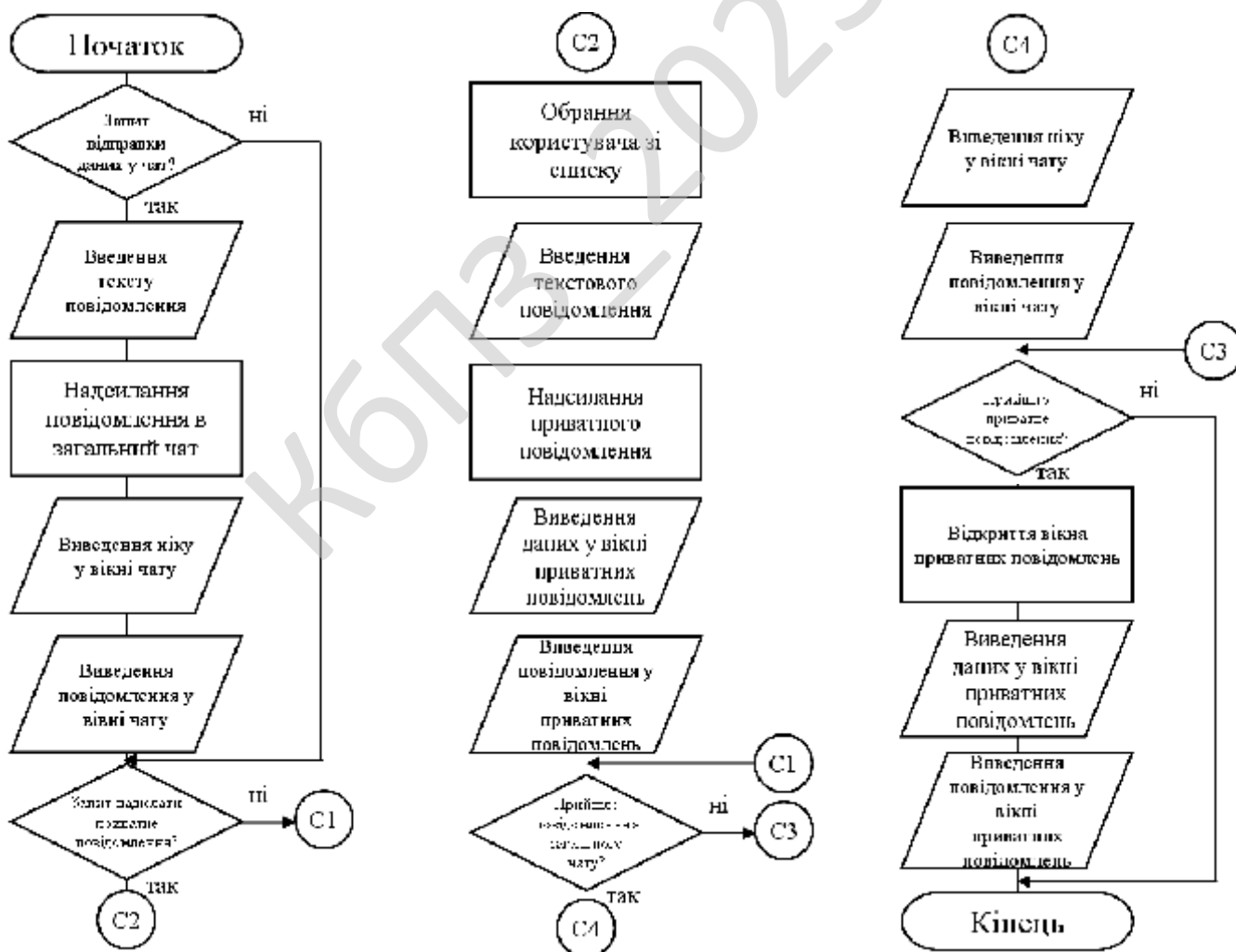


Рисунок 4.2 – Блок схема підпрограми


```

        NE.MaxLength:=510;
// Процедура відправлення повідомлень у приват
        NE.OnKeyPress:=Irc_Channels_MessageText_EditKeyPress;
// Процедура вставки спецтегів
        NE.OnKeyDown:=Irc_Channels_EditKeyDown;
        IRC_Memo:=TIRCMemo.Create(Self);
// вставляємо IRCMemo
        IRC_Memo.Parent:=NP;
        IRC_Memo.Name:=NP.Name+'_IRCMemo';
        IRC_Memo.Align:=alClient;
        IRC_Memo.ReadOnly:=true;
        IRC_Memo.Clear;
// Перехід на Edit при відпусканні миші
        IRC_Memo.OnMouseUp:=Irc_S_MemoMouseUp;
// Смайли
        IRC_Memo.SmilesInfo[0].Smile:=':)';
        IRC_Memo.SmilesInfo[0].ImageIndex:=0;
        IRC_Memo.SmilesInfo[1].Smile:= '=)';
        IRC_Memo.SmilesInfo[1].ImageIndex:=0;
        IRC_Memo.SmilesInfo[2].Smile:=':-)';
        IRC_Memo.SmilesInfo[2].ImageIndex:=0;
        IRC_Memo.SmilesInfo[3].Smile:=':(';
        IRC_Memo.SmilesInfo[3].ImageIndex:=1;
        IRC_Memo.SmilesInfo[4].Smile:=':-(';
        IRC_Memo.SmilesInfo[4].ImageIndex:=1;
        IRC_Memo.SmilesInfo[5].Smile:=':D';
        IRC_Memo.SmilesInfo[5].ImageIndex:=2;
        IRC_Memo.SmilesInfo[6].Smile:=':-D';
        IRC_Memo.SmilesInfo[6].ImageIndex:=2;
        IRC_Memo.SmilesInfo[7].Smile:=':P';
        IRC_Memo.SmilesInfo[7].ImageIndex:=3;
        IRC_Memo.SmilesInfo[8].Smile:=':-P';
        IRC_Memo.SmilesInfo[8].ImageIndex:=3;
        IRC_Memo.SmilesInfo[9].Smile:=':)(';
        IRC_Memo.SmilesInfo[9].ImageIndex:=4;
        IRC_Memo.SmilesInfo[10].Smile:=':;-)';
        IRC_Memo.SmilesInfo[10].ImageIndex:=4;
        IRC_Memo.SmilesInfo[11].Smile:=':8)';
        IRC_Memo.SmilesInfo[11].ImageIndex:=5;
        IRC_Memo.SmilesInfo[12].Smile:=': 8-8-)';
        IRC_Memo.SmilesInfo[12].ImageIndex:=5;
        IRC_Memo.SmilesInfo[13].Smile:=':\';
        IRC_Memo.SmilesInfo[13].ImageIndex:=6;

```

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

```

IRC_Memo.SmilesInfo[14].Smile:='-\'';
IRC_Memo.SmilesInfo[14].ImageIndex:=6;
IRC_Memo.SmilesInfo[15].Smile:='#39+' (';
IRC_Memo.SmilesInfo[15].ImageIndex:=7;
IRC_Memo.SmilesInfo[16].Smile:=':'+#39+' (';
IRC_Memo.SmilesInfo[16].ImageIndex:=7;
IRC_Memo.Smiles:=IRC_ImageList;
// Збільшуємо значення TabSheet_Integer на 1
TabSheet_Integer:=TabSheet_Integer+1;
// Додаємо текст повідомлення
if (UserNick='') or (MessageText='') then
exit
else IRC_Memo.AddFormattedString(TAG_COLOR+'1'+UserNick+'>
                                     '+MessageText);

end;

```

Процедура підключення до сервера.

```

// Підключення
procedure TForm1.IRC_C_ButtonClick(Sender: TObject);
begin
// Очищуємо стан режиму Away
Irc_Away_Status_Label.Visible:=false;
if MyIRC.Connected then
begin
try
MyIRC.Quit(IrcSettingsForm.IrcSettings_Quit_Edit.Text+' -
'+MyIRC.Replies.ClientInfo);
except
// Помилка
on e:Exception do
TIRCMemo(FindComponent(Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddForm
atedString(TAG_COLOR+'4'+E.Message);
end;
MyIRC.Disconnect;
end
else
begin
if (IrcSettingsForm.IrcSettings_Username_Edit.Text='') or
(IrcSettingsForm.IrcSettings_AltNick_Edit.Text='') then
begin
IrcSettingsForm.ShowModal;
exit;
end

```

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

```

else
if Irc_Server_Address_Combobox.Text='' then
begin
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddForm
atedString (TAG_COLOR+'Виберіть будь ласка сервер');
ActiveControl:=Irc_Server_Address_Combobox;
exit;
end
else
if Irc_Nik_Edit.Text='' then
begin
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddForm
atedString (TAG_COLOR+'Введіть будь ласка нік');
ActiveControl:=Irc_Nik_Edit;
exit;
end
else
if Irc_Password_Edit.Text='' then
begin
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddForm
atedString (TAG_COLOR+'Введіть будь ласка пароль');
ActiveControl:=Irc_Password_Edit;
exit;
end
else
Irc_PageControl.ActivePageIndex:=0;
MyIRC.Replies.ClientInfo:=TAG_BOLD+Form1.Caption;
MyIRC.Replies.Version:=MyIRC.Replies.ClientInfo;
MyIRC.Replies.Finger:=IrcSettingsForm.IrcSettings_CTCP_Finger_Edit.Text;
MyIRC.Replies.UserInfo:=IrcSettingsForm.
                IrcSettings_CTCP_Userinfo_Edit.Text;
MyIRC.Nick:=Irc_Nik_Edit.Text;
MyIRC.AltNick:=IrcSettingsForm.IrcSettings_AltNick_Edit.Text;
MyIRC.Username:=IrcSettingsForm.IrcSettings_Username_Edit.Text;
MyIRC.RealName:=IrcSettingsForm.IrcSettings_Username_Edit.Text;
MyIRC.Password:=Irc_Password_Edit.Text;
MyIRC.Host:=Irc_Server_Address_Combobox.Text;
MyIRC.Port:=IrcSettingsForm.IrcSettings_Port_SpinEdit.Value;
try
    MyIRC.Connect (500);
except
    on e:Exception do Reconnect (E.Message);

```

					ВКРМ-122.23.0070.00.00.ПЗ	<i>Арк.</i>
<i>Вим.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		57


```

NewListBox.Name:=NP.Name+'_ListBox';
NewListBox.Width:=120;
NewListBox.Align:=alRight;
NewListBox.Sorted:=true;
NewListBox.ShowHint:=true;
NewListBox.PopupMenu:=Irc_Channel_ListBox_PopupMenu;
// Процедура кліку за списком користувачів на каналі
NewListBox.OnClick:=Irc_Channels_ListBoxClick;
// Процедура подвійного кліку за списком користувачів
NewListBox.OnDbClick:=Irc_Channels_ListBox_PopupMenu_PrivatClick;
// Процедура показу кількості користувачів на каналі
NewListBox.OnMouseMove:=Irc_Channel_ListBoxMouseMove;
// Перехід на Edit при відпусканні миші
NewListBox.OnMouseUp:=Irc_S_MemoMouseUp;
// Вставляємо Splitter
NewSplitter:=TSplitter.Create(Self);
// Вставляємо Splitter
NewSplitter.Parent:=NP;
NewSplitter.Name:=NP.Name+'_Splitter';
NewSplitter.ShowHint:=true;
NewSplitter.Align:=alRight;
NewSplitter.Beveled:=true;
IRC_Memo:=TIRCMemo.Create(Self);
//вставляємо IRCMemo
IRC_Memo.Parent:=NP;
IRC_Memo.Name:=NP.Name+'_IRCMemo';
IRC_Memo.Align:=alClient;
IRC_Memo.ReadOnly:=true;
IRC_Memo.Clear;
// Перехід на Edit при відпусканні миші
IRC_Memo.OnMouseUp:=Irc_S_MemoMouseUp;
// Процедура показу властивостей каналу
IRC_Memo.OnDbClick:=Irc_ChannelSettings_SpeedButtonClick;
// Смайли
IRC_Memo.SmilesInfo[0].Smile:=' :) ';
IRC_Memo.SmilesInfo[0].ImageIndex:=0;
IRC_Memo.SmilesInfo[1].Smile:=' =) ';
IRC_Memo.SmilesInfo[1].ImageIndex:=0;
IRC_Memo.SmilesInfo[2].Smile:=' :-) ';
IRC_Memo.SmilesInfo[2].ImageIndex:=0;
IRC_Memo.SmilesInfo[3].Smile:=' ( ';
IRC_Memo.SmilesInfo[3].ImageIndex:=1;

```

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

```

IRC_Memo.SmilesInfo[4].Smile:=':-(';
IRC_Memo.SmilesInfo[4].ImageIndex:=1;
IRC_Memo.SmilesInfo[5].Smile:=':D';
IRC_Memo.SmilesInfo[5].ImageIndex:=2;
IRC_Memo.SmilesInfo[6].Smile:=':-D';
IRC_Memo.SmilesInfo[6].ImageIndex:=2;
IRC_Memo.SmilesInfo[7].Smile:=':P';
IRC_Memo.SmilesInfo[7].ImageIndex:=3;
IRC_Memo.SmilesInfo[8].Smile:=':-P';
IRC_Memo.SmilesInfo[8].ImageIndex:=3;
IRC_Memo.SmilesInfo[9].Smile:=';)';
IRC_Memo.SmilesInfo[9].ImageIndex:=4;
IRC_Memo.SmilesInfo[10].Smile:=';-)';
IRC_Memo.SmilesInfo[10].ImageIndex:=4;
IRC_Memo.SmilesInfo[11].Smile:='8)';
IRC_Memo.SmilesInfo[11].ImageIndex:=5;
IRC_Memo.SmilesInfo[12].Smile:=' 8-8-)';
IRC_Memo.SmilesInfo[12].ImageIndex:=5;
IRC_Memo.SmilesInfo[13].Smile:=':\';
IRC_Memo.SmilesInfo[13].ImageIndex:=6;
IRC_Memo.SmilesInfo[14].Smile:=':-\';
IRC_Memo.SmilesInfo[14].ImageIndex:=6;
IRC_Memo.SmilesInfo[15].Smile:='#39+'(';
IRC_Memo.SmilesInfo[15].ImageIndex:=7;
IRC_Memo.SmilesInfo[16].Smile:=':'+#39+'(';
IRC_Memo.SmilesInfo[16].ImageIndex:=7;
IRC_Memo.Smiles:=IRC_ImageList;
// Перемикаємося на нову вкладку
  Irc_PageControl.ActivePage:=NP;
// Збільшуємо значення TabSheet_Integer на 1
  TabSheet_Integer:=TabSheet_Integer+1;
// Робота з файлами лога
  IRCLogs(AChannel.Name, '', 'Захід на канал');
end;

```

Розглянемо протокол IRC.

Протокол IRC входить у стандартний набір Інтернет-протоколів та опублікований як RFC, саме як засіб зв'язку декількох чат-серверів працюючих у різних сегментах Інтернету.

Основне достоїнство даного протоколу – можливість зв'язувати між собою декілька територіально розрізнених серверів.

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

Відправлення повідомлення в IRC на рівні протоколу виглядають у такий спосіб: PRIVMSG АДРЕСАТ:текст повідомлення.

Тут АДРЕСАТ може бути ім'ям каналу або ніком. На рисунку 4.3 в центрі розташований світч, розглянемо маршрути пакетів:

- Пакет відправляється клієнтом на світч.
- Зі світча він направляється на сервер.
- На сервері створюється пакет для кожного іншого клієнта й всі вони відправляються на світч.
- Світч розсилає пакети, що прийшли із сервера, по одному кожному клієнту.

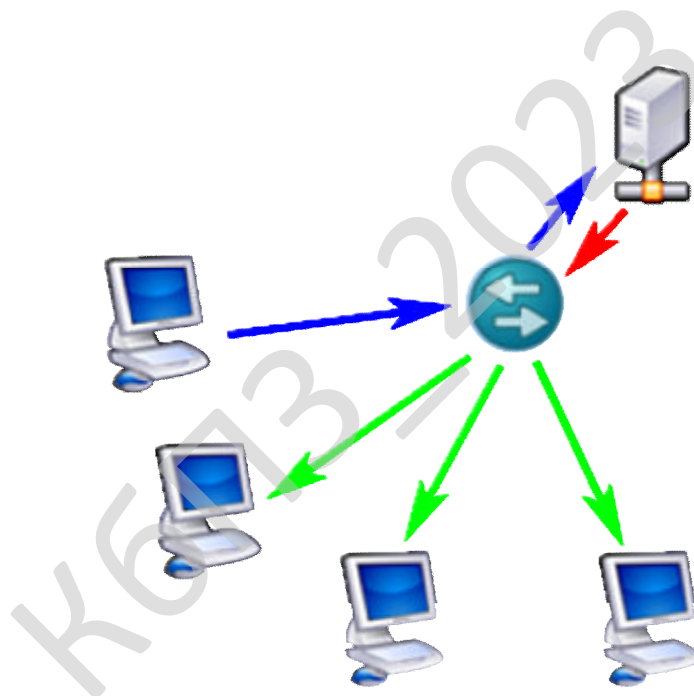


Рисунок 4.3 – Схема проходження пакетів при IRC з'єднанні

IRC-протокол розроблений для систем, що використовують мережний протокол TCP/IP.

IRC являє собою систему телеконференцій, які (через використання моделі клієнт-сервер) добре підходять для роботи на багатьох машинах у розповсюдженій зоні.

Типове налаштування представляє собою процес (сервер), що надає доступ для клієнтів (або інших серверів), і виконуючу доставку/мультиплексування повідомлень та інших функцій.

Оператори

Для підтримки порядку в IRC-мережі, існує спеціальний клас клієнтів (оператори, у даний час їх називають «іркопи»). Хоча, можливості операторів можна розглядати як «небезпечні», вони не підкоряються наказам. Оператори виконують основні мережні завдання, такі як від'єднання й перез'єднання серверів для поліпшення стану мережі або виправлення яких-небудь мережних помилок.

На додаток до можливостей операторів можна додати, що в їхніх силах так само закрити з'єднання між клієнтів і сервером. Правда, подібно це буває тільки при деструктивних і дратівних діях клієнта.

Канали

Канал – це позначення групи з одного або більшого числа клієнтів, які одержують повідомлення, адресовані в цей канал. Канал створюється при з'єднанні першого клієнта з ним і канал зникає, коли його залишає останній клієнт. Поки канал відсутній, будь-які клієнти можуть заволодіти каналом, назвавши свій канал таким же ім'ям.

Імена каналів – рядок (що починається із символу '&' або '#') довжиною до 200 символів. Осторонь від вимог, що перший символ повинен бути або '&' або '#'; обмеження на те, що назва каналу не може містити пробілів (' '), Ctrl-G (^G або ASCII 7), або ком (',' яка використовується для створення списку каналів). Протоколом надається два типи каналів, один розповсюджуваний канал, що відомий всім серверам, приєднаним до мережі.

Ці канали позначені першим символом; доступні тільки тим клієнтам, на сервері яких він існує. Такі канали відрізняються початковим символом '&'.

Для створення нового каналу або входу в існуючий, користувач повинен запросити JOIN каналу. Якщо канал був відсутній, то канал створюється й

користувач, що ввійшов, стає оператором каналу. Якщо канал уже існує, але так чи інакше він не відповідає на спроби ввійти, значить у налаштуваннях каналу встановлений який-небудь із нижчеподаних режимів. Можливо, цей канал тільки-для-запрошених (invite-only), (режим +i), і ви зможете на нього ввійти тільки будучи запрошеним. Користувач може перебувати в декількох каналах одночасно, але рекомендується обмежитися 10-ма каналами.

Якщо в IRC-мережі відбувається розрив, викликаний роз'єднанням двох серверів, канал так само розривається на декілька частин, у яких залишаються користувачі, що сидять на своїх серверах. Коли сервери з'єднуються знову, вони відновлюють частини каналу і його режими. Якщо канал доступний по різні сторони, входження й режими каналу інтерпретуються у своїх манерах.

Оператори каналів

Оператор каналу (також називаються "чоп" або "чаннел оператор") на даному каналі розглядається як власник каналу. На додачу до цього статусу, оператор каналу виконує функції наглядача за порядком на каналі.

Як власник каналу, оператор не засуджується за дії на своїй території, хоча якщо його дії несуть антисоціальні або які інші образливі дії, то буде можливо і потрібно звернутися до IRC-оператора за підтримкою, або навіть зсуву з поста оператора каналу.

Список команд, які можуть використовуватися тільки оператором каналу:

KICK – відключення клієнта від каналу.

MODE – зміна режиму каналу.

INVITE – запрошення клієнта на канал з режимом +i (invite-only).

TOPIC – зміна топіка каналу в режимі каналу – +t.

Оператор каналу ідентифікується символом '@', що слідує за його нікнеймом, щоразу, як він асоціюється з каналом (наприклад, відповіді на команди NAMES, WHO і WHOIS).

до того ж, деякі восьмизначні величини використовуються для контрольних кодів, які розпізнаються як необмежені повідомлення.

Не звертаючи уваги на 8-бітний протокол, необмеженість і змінні цього протоколу найбільш використовуваним терміналом USASCII і телнет-з'єднанням, тому що IRC має скандинавські корні, символи {} розглядаються як еквівалент символів [], але в малому регістрі. Це приводить до критичних результатів, коли визначається унікальність двох нікнеймів.

Повідомлення

Сервери й клієнти створюють повідомлення на які можна відповісти, а можна і не відповідати. Якщо повідомлення містить правильні команди, як описано в попередньому розділі, клієнту варто відповісти, але це не означає, що завжди можна дочекатися відповіді; зв'язок клієнт-сервер і сервер-сервер дуже розсинхронізовані по своїй природі.

Кожне IRC-повідомлення може містити до трьох головних частин: префікс (опціонально), команду й параметри команди (яких може бути до 15). Префікс, команда й всі параметри розділені одним (або більше) символом пробілу (' ', 0x20).

Префікс позначається одним символом, що стоїть спочатку (':', 0x3b), який повинен бути першим символом у повідомленні. Між префіксом і двокрапкою не повинно бути ніяких пробілів. Префікс використовується серверами для позначення джерела появи повідомлення.

Якщо префікс повідомлення загублений, то за джерело повідомлення беруть з'єднання, з якого було отримане повідомлення. Клієнтам не треба користуватися префіксами при відсиланні повідомлення; якщо вони почнуть використовувати префікси, то прийматися будуть тільки правильні й тільки з зареєстрованих нікнеймів.

Якщо вихідні ідентифікатори префіксів не буде знайдені в серверних базах даних, або якщо вони зареєстровані з різних лінків, то сервер буде ігнорувати повідомлення.

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

Команда повинна містити правильну IRC-команду або тризначне число, представлене в ASCII-тексті.

IRC-повідомлення завжди виглядають як рядок символів, що закінчується парою символів CR-LF (Carriage Return – Line Feed. Повернення каретки – переведення рядка) і довжиною рядка, що не перевищує 512 символів (у ці 512 входять і CR-LF). Так що, максимальна довжина рядка для команд і параметрів – 510 символів. Перенос рядка неможливий.

Формат повідомлення в 'псевдо' BNF

Протокол повідомлень повинен бути витягнутий із суміжних потоків октетів. Поточним рішенням стало визначення двох символів, CR і LS, як роздільників повідомлень. Ігнорування порожніх повідомлень, які використовують послідовності CR-LF між повідомленнями без яких-небудь проблем.

Розпаковане повідомлення перевіряється усередині компонентів <prefix>, <command> і список параметрів підрівнюється за допомогою <middle> або <trailing> компонентами.

```
<message> ::= [ ':' <prefix> <SPACE> ] <command> <params> <crlf>
<prefix>   ::= <servername> | <nick> [ '!' <user> ] [ '@' <host> ]
<command>  ::= <letter> { <letter> } | <number> <number> <number>
<SPACE>    ::= ' ' { ' ' }
<params>   ::= <SPACE> [ ':' <trailing> | <middle> <params> ]
<middle>   ::= <Будь-яка *не порожня* послідовність октетів, що не
включає в себе пробіл, або NUL, або CR, або LF; першою не може бути ':'>
<trailing> ::= <Будь-яка, можливо *порожня*, послідовність октетів, що не
включає в себе NUL або CR, або LF>
<crlf>     ::= CR LF
```

Зауваження:

1. <SPACE> містить тільки символ(и) пробілу (0x20). Табуляцію й інші контрольні символи розглядаються як НЕ-ПОРОЖНІ-ПРОБІЛИ (NON-WHITE-SPACE).

2. Після добування списку параметрів, всі параметри рівняються за допомогою <middle> або <trailing>.

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

префікс відправника, тризначний номер і ціль відповіді. Нумерація відповідей не допускає повідомлення від клієнта-відправника; будь-які такі повідомлення, отримані сервером, віддаляються. У всіх інших випадках, нумерований відповідь просто звичайне повідомлення, що містить змінну. Змінна має вигляд тризначного номера, що незмінно краще, ніж рядок символів.

Встановлення з'єднання сервер-клієнт

При з'єднанні з IRC-сервером, клієнту посилається MOTD (якщо є присутнім), а так само поточне число серверів/клієнтів (подібно команді LUSER). Так само сервер вимагає повідомлення до клієнта, що містить ім'я й версію, як і будь-які інші повідомлення-вступи.

Після поділу із цим, сервер повинен відправити назовні новий нікнейм користувача й іншої інформації, як постачання себе (команда USER) і як відкриваючий сервер (з DNS/сервери ідентифікації). Сервер повинен послати цю інформацію з NICK першим наступної USER.

Встановлення з'єднання сервер-сервер

Процес установки з'єднання сервер-до-сервера чреватий небезпекою, так як багато можливих областей, де виникають проблеми – найменші з яких – конфліктні ситуації. Після з'єднання сервера, слідуючи PASS/SERVER, якими при з'єднанні, сервера обмінюються парами PASS/SERVER, за допомогою яких встановлюються паролльні лінки. За допомогою цих же команд і їх відповідей, сервера обмінюються інформацією про з'єднання, що описана нижче. При ініціалізації серверних запитів пари PASS/SERVER, так само перевіряється, що сервер належним чином відповідає на ідентифікацію перед підтвердженням з'єднання (перевіряється для того, щоб переконатися в тому, що це дійсно сервер).

Розрив з'єднання сервер-клієнт

Коли клієнт закриває з'єднання, створюється QUIT-повідомлення від імені клієнта сервером, до якого приєднаний клієнт. Не використовується й не створюється ніяке інше повідомлення.

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

Розрив з'єднання сервер-сервер

При закритті з'єднання сервер-сервер, і той й інший віддалено створюють SQUIT або 'натуральні' причини закриття з'єднання з IRC-мережею, які повинні бути відомі всім серверам. Для цього сервер відсилає список SQUIT'ів (по одному на кожний сервер, що має з'єднання з ним) і список QUIT'ів (знову, по одному для кожного клієнта, з'єднаних з ним).

4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм LOKI_91. Механізм алгоритму LOKI_91 подібний DES (рисунок 4.5). Блок даних розщеплюється на ліву й праву половини й проходить 16 раундів, що досить нагадує DES. У кожному раунді права половина спочатку піддається операції XOR із частиною ключа, а потім розширювальній перестановці (таблиця 4.1).

Таблиця 4.1 – Перестановка з розширенням

4	3	2	1	32	31	30	29	28	27	26	25
28	27	26	25	24	23	22	21	20	19	18	17
20	19	18	17	16	15	14	13	12	11	10	9
12	11	10	9	8	7	6	5	4	3	2	1

48-бітовий вихід розділяється на чотири 12-бітових блоки. У кожному блоці виконується така підстановка з використанням S-блоку: береться кожний 12-бітовий вхід, 2 старших і 2 молодших біти використовуються для утворення номера r , а вісім внутрішніх біт утворюють номер c . Вихід S-блоку, O , має наступне значення: $O(r,c) = (c + ((r*17) \oplus 0xff) \& 0xff)^{31} \bmod P_r$.

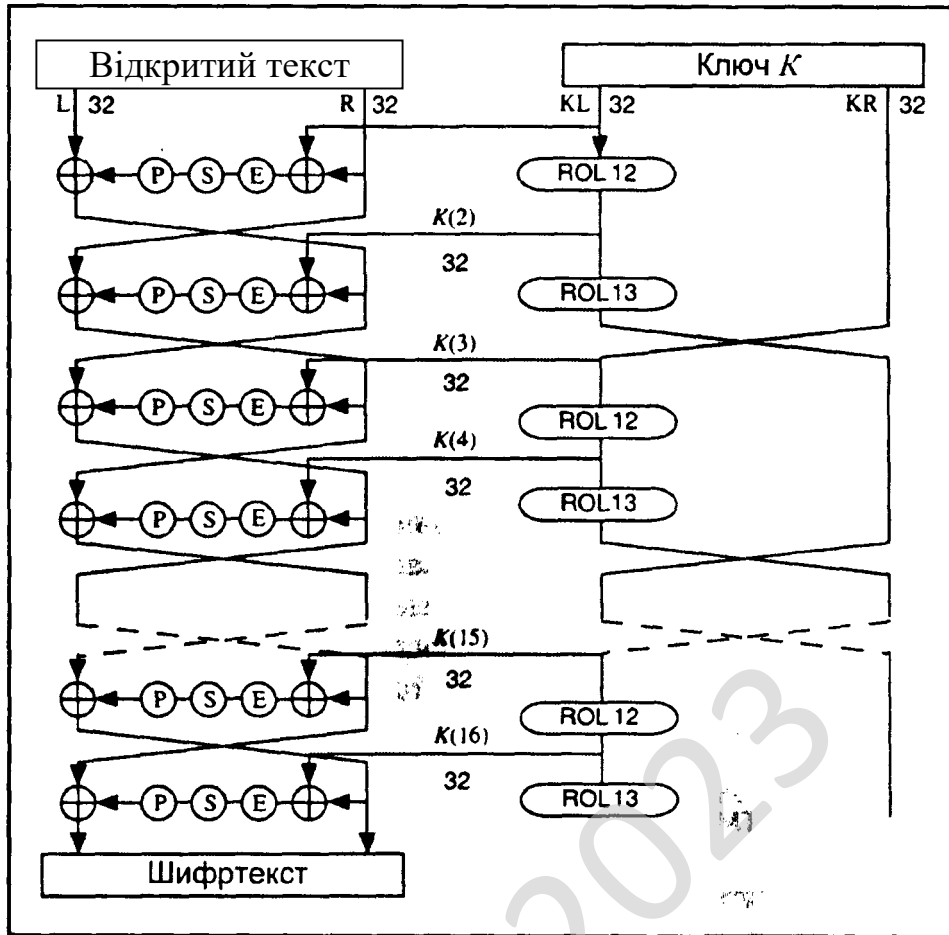


Рисунок 4.5 – Алгоритм LOKI_91

Таблиця 4.2 – Значення P_r

r	1	2	3	4	5	6	7	8
P_r	375	379	391	395	397	415	419	425
r	9	10	11	12	13	14	15	16
P_r	433	445	451	463	471	477	487	499

Після цього чотири 8-бітових результати знову поєднуються, утворюючи 32-бітове число, що піддається операції перестановки, описаній в таблиці 3. Нарешті, для одержання нової лівої половини виконується операція XOR правої половини з колишньою лівою половиною, а ліва половина стає новою правою

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розглянемо розроблене ПЗ яке зображено на рисунку 5.1. Програма має простий та інтуїтивно зрозумілий інтерфейс. Для початку роботи слід здійснити авторизацію: ввести логін, пароль та назву серверу IRC-чату. Після цього в полі повідомлень чату з'явиться лог підключення та довідкова інформація про обраний сервер.

Інтерфейс програми надає доступ до наступних можливостей: Вибір та підключення до IRC-сервера; Перегляд списку наявних каналів та підключення до вибраного каналу; Додавання підключених каналів в «Обране»; Перегляд властивостей каналу; Очищення текстового поля повідомлень чату; Перегляд списку команд та запитів, запуск вибраних команд, запитів; Налаштування та використання IRC-бота; Ведення та перегляд архіву повідомлень; Ведення та перегляд лог-файлу; Налаштування параметрів з'єднання та вибір режимів роботи IRC-чату; Перегляд довідки.

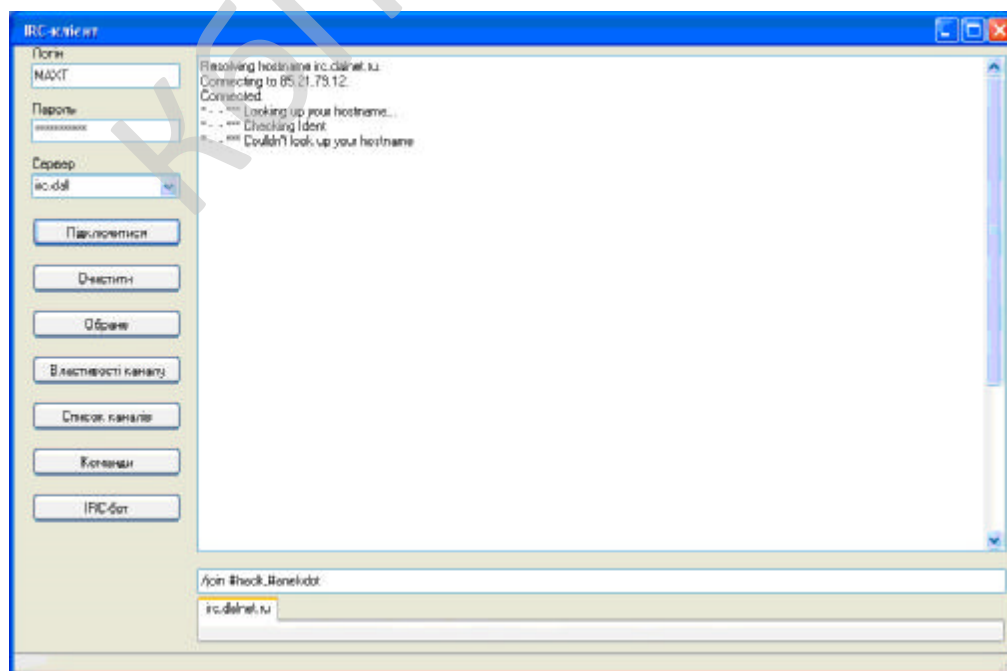


Рисунок 5.1 – Головне вікно програми (підключення до сервера)

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

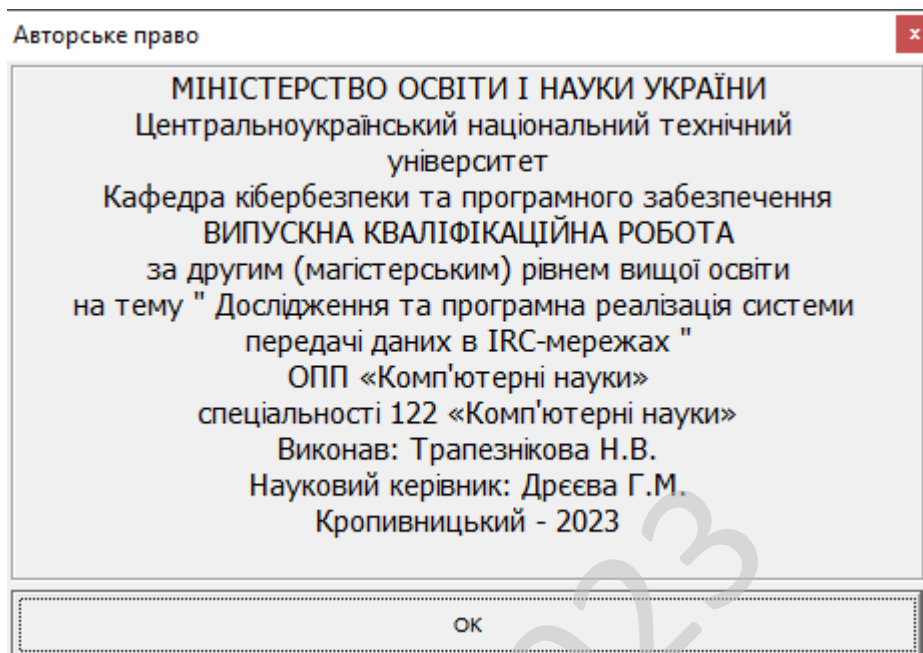


Рисунок 5.2 – Авторське право

Обрано умови розповсюдження – Freeware. Це власницьке програмне забезпечення, котре можна Безоплатно використовувати протягом необмеженого терміну без обмежень у функціональності, і поширюване без сирцевих кодів. Дуже часто плутають поняття «безплатне програмне забезпечення» та «вільне програмне забезпечення», хоча вони суттєво відрізняються.

Безплатне програмне забезпечення можна безоплатно встановлювати та використовувати (іноді з певними обмеженнями, як, наприклад, «безплатне для домашнього або некомерційного вжитку»), в той час як вільне програмне забезпечення можна продавати за будь-яку суму, але при тому, у користувача, котрий його отримує, повинні бути права на вивчення, модифікацію та поширення сирцевих кодів одержаної програми.

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи передачі даних в IRC-мережах.

Метою розробки є дослідження та програмна реалізація системи передачі даних в IRC-мережах.

Об'єктом дослідження є процес передачі даних в IRC-мережах.

Предметом дослідження є методи передачі даних в IRC-мережах.

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

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

- Удосконалено метод передачі даних в IRC-мережах.
- Розроблено вітчизняний продукт передачі даних в IRC-мережах, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

7 ДАНІ ПРО ЕКОНОМІЧНУ ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ

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

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

В магістерській роботі було проведено дослідження та виконана програмна реалізація системи передачі даних в IRC-мережах.

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

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

Таблиця 7.1 – Початкові дані

Показники	Позначення	Характеристика або величина
1	2	3
1. Кількість розроблених програм період, шт.	N	1
2. Кількість екземплярів програм, шт.	Ne	19
3. Запланований термін розробки, днів	Fpq	60 (3 місяці)
4. Група задачі підсистеми управління (1-6)	–	1
5. Ступінь новизни задачі (А, Б, В, Г)	–	Б
6. Складність алгоритму (1, 2, 3)	–	2

Продовження таблиці 7.1

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

Продовження таблиці 7.1

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

7.2 Розрахунок трудомісткості розробки програмної продукції

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

Визначимо трудомісткість розробки ПЗ для стадії РП.

Обчислюємо номінальні трудовитрати, люд-міс.:

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

де: A – коефіцієнт Боема, $A = 2,45$;

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

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

B – показник ступеня, що визначається співвідношенням:

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

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

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

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

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

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

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

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

Ці коефіцієнти дозволяють диференційовано оцінювати результати роботи програмістів, беручи до уваги швидкодію програми, використання різноманітних обчислювальних платформ і інструментів розробки, взаємодію декількох серверів, вимоги до об'ємів баз даних і ін.

Визначаємо підсумкові трудовитрати по стадії робочий проект, люд-дні:

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

де: C – визначений емпірично коефіцієнт, запропонований авторами методики, (МВ, додаток 4). Для вибраної мови програмування він складає 2,66;

S – коефіцієнт стиснення (або подовження) графіка робіт %, що дозволяє коректувати терміни розробки ПЗ згідно встановленим вимогам. Вибираємо в межах (25...350)%. Приймаємо $S = 100$ %

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

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

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

Таблиця 7.2 – Визначення трудомісткості розробки програмного забезпечення

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

7.3 Визначення чисельності виконавців і планового фонду зарплати

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

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

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

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

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

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

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

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

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

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

Виходячи з цього фонд робочого часу інженерів-електронщиків складає:

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

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

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

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

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

Для забезпечення нормального технічного обслуговування засобів ТО та мереж, необхідно прийняти найбільше ціле значення розрахункової чисельності інженерів-електронщиків.

Чисельність інженерів-системотехніків, адміністраторів мережі, дизайнерів WEB вузлів, системних програмістів (аналітиків), бухгалтерів-економістів визначається за потребою в залежності від функціональних обов'язків. Після визначення чисельності персоналу складається штатний розклад.

Таблиця 7.4 – Розрахунок чисельності штатного персоналу сектору системного та адміністративного обслуговування засобів ОТ та комп'ютерних мереж

Посада	Вид роботи	Час	К-ть штатних одиниць
Адміністратор загальної мережі, аналітик	Адміністрування локальної мережі, поштового та серверу DNS (OC FreeBSD), маршрутизатора Cisco, доменного контролеру Windows Server 2019, серверу доступу ADSL (OC Linux), налаштування ADSL, VPN PPPoE, Frame Relay, Wi-Fi	2	0,5
	Налаштування і конфігурування базової станції безпроводного зв'язку (CMTS)	0,5	
	Розробка та впровадження проектів з організації зв'язку між віддаленими об'єктами, ЛОМ	0,5	
	Забезпечення цілодобової роботи зв'язку клієнтів до мережі Інтернет	1	
Всього		4	

Продовження таблиці 7.4

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

Складемо штатний розклад виконавців.

Таблиця 7.5 – Штатний розклад виконавців

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

Розрахуємо середньоденну зарплату одного виконавця:

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

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

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

7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника

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

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

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

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

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

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

Згідно даних ТОВ науково-дослідницького консалтингового підприємства «Пектораль» (м. Кіровоград, вул. Глинки 16) ціна одного квадратного метра площі новобудови, вік якої не перевищує 25 років, по місту складає 500...1600 у.о./ m^2 . Враховуючи, що курс складає 1 у.о. = 38 грн. приймаємо для розрахунку вартість одного метра квадратного рівною 20000 грн./ m^2 . На кожне робоче місце у середньому потрібно 8 m^2 . З урахуванням цього:

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

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

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

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

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

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

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

Специфікація на обчислювальну техніку наведена в таблиці 7.7.

Дані по оптовій ціні на обладнання та комплектуючі вибирались за прайсом фірми Комп'ютерторг за 24.10.23 – джерело <http://computorg.ua/price.html>

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Персональний комп'ютер		10947
Системний блок		7347
Процесор	Intel Pentium G6405 1200, 2 ядра, 4 потоки, 4.1 GHz, TDP - 58 Вт, 14nm, 8 GT/s, BOX	-
Системна плата	GIGABYTE H510M S2H V3 1200, 3200 MHz, LAN - 1 Гбит/с, D-Sub (VGA), DisplayPort, HDMI, 1 x M.2 2280, 4 x SATA 6.0 Gb/s, Micro-ATX	-
Відеокарта	Intel UHD Graphics 610	-
Жорсткий диск	SD 2.5" 256GB Mibrand, 3D TLC NAND, 2.5", SATA III (6Gb/s)	-
Оперативна пам'ять	DDR4 16GB 3200 MHz Dato (DT8G4DLDND32) DDR4, 8 ГБ, В наборі - 2	-
DVD-привод	DVDRW Pioneer DVR-TD10RS SATA Slim Black Bulk (DVR-TD10RS)	-
Корпус	ATX Middle Tower FOXCONN Pro, 3GTLA-489, PSU 550W(FSP Brand: ATX-350PNR, 12cm), black, (front bezel – black+light silver; body material – 0.6mm), 80mm fan (rear), 2xUSB2.0/AUDIO/MIC, Air Duct, Tool-less chassis design, Thermally Advantaged Chassis	-
Клавіатура	Vinga KB110BK для ПК, мембранна, провідна, USB, повно розмірна, англійська, українська, Клавіш - 104, вологостійка, Шнур - 1.5 м, 440 x 150 x 22 мм, 0.384 кг	-

Продовження таблиці 7.6

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Кулер	–	–
Кардрідер внутрішній	USB 2.0 Card reader STORM CR-35U1A4-B, int. 3.5", 1*USB2.0+AUDIO+1394, multi: All Type Cards, black	-
Маніпулятор	Мишка Defender Optimum MB-270 Black	-
Монітор	22" TFT, ASUS VW223D (5ms, 300/3000: 1, 170/160, D-SUB, Wide)	3600
Принтер лазерний	Canon i-SENSYS LBP6030W	2700
Принтер струминний	Epson Stylus Photo P50 (C11CA45341) + USB cable	5500
Копіювальний апарат	Canon i-SENSYS MF217W with Wi-Fi	5965

Таблиця 7.7 – Балансова вартість обчислювальної техніки

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

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

Для визначення необхідної кількості капітальних вкладень складемо таблицю 7.8.

Таблиця 7.8 – Вартість основних фондів та амортизаційні відрахування розробника

Групи та види основних фондів	Балансова вартість, грн.	Амортизація	
		Норма, %	Відрахування, грн.
1	2	3	4
Група 3			
1. Будівлі	1280000	-	-
2. Передавальні пристрої	128000	-	-
Всього по групі	1408000	5	70400
Група 4			
3. Обчислювальна техніка	114885	-	-
Всього по групі	114885	50	57442,5
Група 5, 6			
4. Вимірювальні пристрої	5190	25	1297,5
5. Транспортні засоби	97500	20	19500
6. Господарський інвентар	28000	25	7000
Всього по групі	130690	-	27797,5
7. Нематеріальні активи	19000	10	1900
Разом	$K_p = 1672575$		$A_p = 157540$

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

7.5 Визначення собівартості розробки та ціни програмної продукції

Визначимо основну зарплату виконавців:

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

де: N_e – кількість екземплярів програм, шт.

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

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

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

де: H_q – норматив додаткової зарплати, %.

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

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

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

де: H_c – відрахування на соціальні потреби, %.

$$C_{oc} = 0,01 \cdot 22(6600 + 660) = 1597 \text{ грн.}$$

Визначимо загальногосподарські витрати (електроенергію, ремонт і утримання приміщень і т.д) за нормативом $H_z = 15\%$ від основної зарплати:

$$G_{ocn} = Z_o \cdot H_z \cdot 0,01, \quad (7.14)$$

де: H_z – загальногосподарські витрати, %.

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

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

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		89

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

де: Z_{M1} – вартість паперу, грн.;

Z_{M2} – вартість запам'ятовуючих пристроїв, грн.;

Z_{M3} – вартість фарби, картриджів, тонеру, грн.;

N_e – кількість екземплярів програм, шт.

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

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

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

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

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

де: $Ц_\delta$ – вартість дисків CD/DVD: CDR TDK 700Mb, 80Min, 52x Cake box – 25 грн./шт., DVD-R LG 4,7Gb, 16x speed Cake box – 57 грн./шт.

$$Z_{M2} = 25 \cdot 2 + 57 = 107 \text{ грн.}$$

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

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

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

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

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

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

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

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

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

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

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

Таблиця 7.9 – Нормативна калькуляція собівартості розробки програмного забезпечення задачі

Найменування статей витрат	Позначення	Величина, грн
1	2	3
1. Основна зарплата виконавців	Z_o	6600
2. Додаткова зарплата виконавців	Z_d	660
3. Відрахування на соціальні потреби	C_{oc}	1597
4. Загальногосподарські витрати	Γ_{ocn}	990
5. Витрати на матеріали	Z_M	101
6. Освоєння нових операційних систем, мов програмування	O_n	990
7. Амортизація основних фондів	A_m	2073
8. Повна собівартість програмного забезпечення	C_n	13011
9. Плановий прибуток	Π_p	6505,5
10. Ціна підприємства $C_n = C_n + \Pi_p$	C_n	19516,5
11. Податок на додану вартість $\Pi_{ДВ} = 0,01 \cdot N_{об} \cdot C_n$	$\Pi_{ДВ}$	3903,3
12. Відпускна ціна програмної продукції $C = C_n + \Pi_{ДВ}$	C	23419,8

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

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

де: A_p – загальна річна сума амортизаційних відрахувань, грн.

$$A_m = 157540 \cdot 3 / (19 \cdot 12) = 2073 \text{ грн.}$$

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

$$C_n = Z_o + Z_d + C_{oc} + \Gamma_{ocn} + Z_m + O_n + A_m. \quad (7.21)$$

$$C_n = 6600 + 660 + 1597 + 990 + 101 + 990 + 2073 = 13011 \text{ грн.}$$

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

Для даного програмного забезпечення рівень рентабельності складає 50%.

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

де: P_n – рівень рентабельності, %.

$$P_p = 0,01 \cdot 50 \cdot 13011 = 6505,5 \text{ грн.}$$

7.6 Визначення об'єму капітальних вкладень у споживача програмної продукції

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

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

Найменування капітальних вкладень	Сума за варіантами, грн.	
	Базовий	Новий
Вартість програмної продукції	–	23420
Всього капітальних витрат	–	23420

7.7 Визначення експлуатаційних витрат

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

Таблиця 7.11 – Розрахунок експлуатаційних витрат у споживача програмної продукції

Найменування статей витрат	Позначення	Сума витрат за варіантами, грн.	
		Базовий	Новий
1. Витрати на налаштування та технічне обслуговування	Z_p	32208	14091
2. Витрати на електроенергію	$Z_{ел}$	0	0
3. Витрати на амортизацію	$Z_{ам}$	0	5855
Всього витрат за рік	I	32208	19946

Витрати на налаштування та технічне обслуговування системи:

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

де: T_p – кількість годин налаштування та обслуговування за рік, год.;

Z_z – заробітна плата персоналу, грн / год.

Після купівлі нового програмного забезпечення кількість годин на проведення робіт по налаштуванню і обслуговуванню системи зменшилася з 240 до 105 годин на рік, тому витрати на технічне обслуговування зменшилися з:

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

до:

$$Z_{p \text{ нов}} = 105 \cdot 100 \cdot 1,1 \cdot 1,22 = 14091 \text{ грн}.$$

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

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

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

$Z_{ел\ баз} = Z_{ел\ нов}$ При впровадженні програмного забезпечення витрати на електроенергію не змінилися

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

Таблиця 7.12 – Розрахунок амортизаційних відрахувань

Групи основних фондів	Норма амортизації %	Балансова вартість, грн., за варіантами		Сума відрахувань, грн за варіантами	
		Базовий	Новий	Базовий	Новий
Програмна продукція	25	–	23420	–	5855
Всього відрахувань	-	–	23420	–	5855

7.8 Визначення економічної ефективності програмної продукції

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

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

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

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

$$E_e = (21150 - 14100) \cdot 19 - (0,05 \cdot 1408000 + 0,5 \cdot 114885 + 0,25 \cdot 33190 + 0,2 \cdot 97500 + 0,1 \cdot 19000) \cdot 3/12 = 94565 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у виробника програмної продукції:

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

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

$$T_e = \frac{1672575}{(21150 - 14100) \cdot 19 \cdot 12 / 3} = 3,1 \text{ років.}$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1. Кількість екземплярів програми	Прим.	19
2. Повна собівартість розробленої програми	Грн.	14100
3. Ціна розробленої програми	Грн.	21150
4. Плановий прибуток від реалізації розробленої програми	Грн.	7050
5. Рентабельність програмної продукції	%	50
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	1672575
7. Загальний прибуток від реалізації програмної продукції	Грн.	133950
8. Величина економічного ефекту при виготовлені програмної продукції	Грн.	94565
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років.	3,1
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	23420
11. Величина економічного ефекту у користувача програмної продукції	Грн.	6407
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	1,9

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

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

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

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

$$E_{cn} = (32208 - 19946) - 0,25 \cdot 23420 = 6407 \text{ грн.}$$

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

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

$$T_{cn} = \frac{23420}{32208 - 19946} = 1,9 \text{ року.}$$

7.9 Висновки

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

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		96

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Охорона праці – система збереження життя і здоров'я працівників у процесі трудової діяльності, що включає правові, соціально-економічні, організаційні, технічні, санітарно-гігієнічні, лікувально-профілактичні, реабілітаційні та інші заходи. Загальні положення державної політики, щодо галузі охорони праці зазначені у Законі України “Про охорону праці”. Цей Закон визначає основні положення щодо реалізації конституційного права працівників на охорону їх життя і здоров'я у процесі трудової діяльності, на належні, безпечні і здорові умови праці, регулює за участю відповідних органів державної влади відносини між роботодавцем і працівником з питань безпеки, гігієни праці та виробничого середовища і встановлює єдиний порядок організації охорони праці в Україні [20]. Законодавство про працю містить норми і вимоги з техніки безпеки і виробничої санітарії, норми, що регулюють робочий час і час відпочинку, звільнення та переведення на іншу роботу, норми праці щодо жінок, молоді, гігієнічні норми і правила тощо. Загальний нагляд за додержанням норм охорони праці покладено на прокуратуру, спеціальний – на професійні спілки. Контроль за безпекою праці здійснюють також, державні й відомчі спеціалізовані інспекції. Науково-технічний прогрес вніс серйозні зміни в умови виробничої діяльності робітників розумової діяльності. Їх праця стала більш інтенсивною, напруженою і вимагає значних витрат розумової, емоційної і фізичної енергії. Це призвело до необхідності у знаходженні комплексного рішення проблем ергономіки, гігієни і організації праці, регламентації режимів праці та відпочинку. Охорона здоров'я робітників, забезпечення безпеки умов праці, ліквідація та профілактика професійних захворювань і виробничого травматизму складає одну з головних турбот людського суспільства.

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

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

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

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

8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста

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

Таблиця 8.1 – Розміри приміщення

Найменування	Значення, м
Ширина	5,38
Довжина	5,95
Висота	2,8

Таблиця 8.2 – Площа та обсяг приміщення, на одного працюючого*

Геометрична характеристика	Одиниця виміру	Нормативне значення*	Фактичне значення
Площа, S	м ²	не менше 6.0	8
Об'єм, V	м ³	не менше 20.0	22,4

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

У зазначеному приміщенні працюють четверо людей. За даними, які наведено у табл. 8.1, та табл. 8.2, можна зробити висновок, що площа та об'єм приміщення у розрахунку на одно робоче місце програміста не відповідають нормативним вимогам ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» [5], але відповідають нормативним вимогам Наказу Міністерства соціальної політики України № 207, від 14.02.2018 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» [5] та НПАОП 0.00-1.28-10 «Правила охорони

праці під час експлуатації електронно-обчислювальних машин»). Таним чином можна зробити висновок, що санітарно-гігієнічні умови праці на робочому місці програміста відповідають вимогам.

Температура повітря в приміщенні визначається впливом температури зовнішнього повітря і тепловою енергією, яка виділяється всередині приміщення. Джерелами виділення теплоти в даному приміщенні є електроустаткування, освітлювальні прилади, а також люди. У світлий час доби джерелом надлишкового тепла є сонячна радіація. Згідно Постанови № 42 від 01.12.1999 Головного державного санітарного лікаря України, робота, виконувана в даному приміщенні, відноситься до категорії Іа. В цьому випадку людина витрачає енергії до 120 ккал у годину. Вологість повітря в приміщенні визначається впливом багатьох факторів, серед яких: вологість атмосферного повітря, виділення вологи людьми (при диханні та випарами з поверхні шкіри).

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

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

Таблиця 8.3 – Оптимальні і фактичні значення параметрів мікроклімату

Пора року	Оптимальні для Іа			Фактичні		
	Температура, °С	Вологість, %	Швидкість повітря, м/с	Температура, °С	Вологість, %	Швидкість повітря, м/с
Холодна	22-24	40-60	0,1	22-23	40-55	0,1
Тепла	23-25	50-70	0,1	24-25	50-65	0,11

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

В літню пору застосовується кондиціонер.

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

У приміщенні знаходяться наступні джерела шуму: принтер HP 1100, електродвигуни вентиляторів ЕОМ.

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

З 2019 року діють Державні будівельні норми України “Природне і штучне освітлення” – ДБН В.2.5-28:2018 [1], у яких прописані вимоги до використання всіх освітлювальних приладів, у т.ч. світлодіодних.

Працю працівника, який постійно працює за комп'ютером, згідно ДБН В.2.5-28:2018 [1], можна віднести до роботи з малою точністю (найменший розмір об'єкта розрізнення від 1 до 5 мм) V-го розряду зорової роботи, з великою контрастністю об'єкта розрізнення (символів на екрані дисплея), з темним тлом (під розряд зорової роботи В). Приміщення можна віднести до 1-ої групи приміщень, у яких проводиться розрізнення об'єктів зорової роботи при фіксованому напрямку лінії зору того, що працює на робочу поверхню. Для такого типу приміщень і розряду зорової роботи нормоване значення коефіцієнта природної освітленості (КПО) робочої поверхні (при поєднаному, спільному освітленні), повинен становити не більше 1,5%, освітленість при штучному висвітленні повинна становити 300 Лк. [1], Крім того все поле зору повинне бути освітлено достатньо рівномірно – ця основна гігієнічна вимога. Так як яскраве світло на ділянці периферійного зору значно збільшує напруженість очей і, як наслідок, призводить до їх швидкої стомлюваності, ступінь освітлення приміщення і яскравість екрану комп'ютера повинні бути приблизно однаковими.

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		101

8.4 Розробка заходів з умов поліпшення охорони праці

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

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

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

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

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

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

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		102

В приміщенні (де відсутні джерела виділення шкідливих речовин) працює одна людина. Робота пов'язана з використанням ПЕОМ. Розміри приміщення: $A = 4$ м, $B = 3,5$ м, $H = 2,8$ м, устаткування займає 15% об'єму. Визначити найменшу необхідну кількість повітря для вентиляції.

Для приміщень, в яких відсутні виділення шкідливих речовин у повітрі, розрахунок вентиляції здійснюється залежно від кількості працюючих.

Необхідна кількість повітря (м^3 /год.), яка забезпечує відповідність параметрів повітря робочої зони нормованим значенням, визначається за наступною формулою:

$$L = L' \cdot N, \quad (8.1)$$

де L' – нормативна кількість повітря на одного працюючого, яка залежить від питомого об'єму приміщення, м^3 / (год.-люд.);

N – кількість працюючих.

Питомий об'єм приміщення V_p , (м^3 /люд.), визначається за формулою:

$$V_p = V / N, \quad (8.2)$$

де V – об'єм приміщення, м^3 .

Визначаємо вільний об'єм приміщення

$$V = A \cdot B \cdot H \cdot 0,85 = 4 \cdot 3,5 \cdot 2,8 \cdot 0,85 = 33,3 \text{ м}^3.$$

Питомий вільний об'єм складає

$$V' = V / N = 33,2 / 1 = 33,2 \text{ м}^3 / \text{люд.} < 20 \text{ м}^3 / \text{люд.}$$

Нормована кількість повітря на одну людину при $V' < 20 \text{ м}^3 / \text{люд.}$ становить $30 \text{ м}^3 / (\text{год.} \cdot \text{люд.})$.

8.5 Висновки до розділу

У даному розділі магістерської роботи проведено аналіз умов працівника робота якого пов'язана з комп'ютерною технікою. Проведено аналіз основних санітарно – гігієнічних показників в заданому приміщенні, де працівник зайнятий постійною роботою за комп'ютером.. Створені умови повинні забезпечувати

комфортну роботу. На підставі вивченої літератури з даної проблеми, були зазначені оптимальні параметри мікроклімату, освітлення, допустимі рівні шуму та іонізуючого випромінювання при роботі з ПЕОМ, а також розраховано найменшу необхідну кількість повітря для вентиляції.

Дотримання умов, що визначають оптимальну організацію робочих місць працівників, дозволить зберегти гарну працездатність протягом усього робочого дня, підвищить як в кількісному, так і в якісному відносінах продуктивність їх праці.

Список використаних джерел інформації

1. Наказ Міністерства соціальної політики України 14.02.2018 № 207 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/z0508> (дата звернення 19.10.22).

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

3. Наказ Міністерства внутрішніх справ України 30.12.2014 №1417 «Про затвердження Правил пожежної безпеки України» – Режим доступу до ресурсу <https://zakon.rada.gov.ua/laws/show/z0252-15#Text> (дата звернення 19.10.22).

4. Наказ Міністерства освіти і науки України 15.08.2016 № 974 «Про затвердження Правил пожежної безпеки для навчальних закладів та установ системи освіти України» – Режим доступу до ресурсу <https://zakon.rada.gov.ua/laws/show/z1229-16#Text> (дата звернення 19.10.22).

5. Постанова № 42 від 01.12.1999 Головного державного санітарного лікаря України «Санітарні норми мікроклімату виробничих приміщень ДСН 3.3.6.042-99 – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/va042282-99> (дата звернення 19.10.22).

6. Державні будівельні норми України: ДБН В.2.5-28:2018. – Режим доступу до ресурсу: <https://goo.su/9AkQ> (дата звернення 19.10.22).

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		104

7. Методичні рекомендації до виконання розділу "Заходи з охорони праці та техніки безпеки" випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти для здобувачів вищої освіти спеціальностей 123 "Комп'ютерна інженерія" та 122 "Комп'ютерні науки" / М-во освіти і науки України, Центральноукраїн. нац. техн. ун-т, каф. кібербезпеки та програм. забезпечення; [укл. О.В. Оришака, К.М. Марченко]. – Кропивницький: ЦНТУ, 2022. – 19 с. [Електронний ресурс]. – Режим доступу : <http://dspace.kntu.kr.ua/jspui/handle/123456789/12240> (дата звернення 19.10.22).

КБПЗ – 2023

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		105

9 ОСНОВНІ ВИСНОВКИ

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

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти наведені теоретичне узагальнення й рішення наукового завдання дослідження методів передачі даних в IRC-мережах.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем передачі даних в IRC-мережах.
- Досліджена система передачі даних в IRC-мережах.
- На основі отриманих результатів досліджень створена програмна реалізація системи передачі даних в IRC-мережах.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання передачі даних в IRC-мережах.

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

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

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

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		106

Програма реалізована на мові високого рівня Delphi 10.4 Sydney. Дана мова програмування дозволяє найбільш ефективно обробляти дані. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

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

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм LOKI_91.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

Розроблена програма має реальний економічний ефект від її впровадження у виробництво у сумі 6407 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 1,9 роки.

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		107

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Трапезнікова Н.В. Дослідження та програмна реалізація системи передачі даних в IRC-мережах // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2023.
2. Ranjan Parekh. Fundamentals of Image, Audio, and Video Processing Using MATLAB® With Applications to Pattern Recognition. CRC Press. 2021. 406 p.
3. Alasdair McAndrew. A Computational Introduction to Digital Image Processing. Chapman & Hall. 2021. 560 p.
4. Peter Shirley, Steve Marschner. Fundamentals of Computer Graphics. 2009
5. Михайло Пічугін, Іван Канкін, Володимир Воротніков Комп'ютерна графіка. Навчальний посібник / Центр навчальної літератури 346 с. 2019р.
6. Маценко В.Г. Комп'ютерна графіка: Навчальний посібник. – Чернівці: Рута, 2009 – 343 с.
7. Інженерна комп'ютерна графіка: підручник / В.В. Проців [та ін.] / М-во освіти і науки України, Нац. гірн. унт-т. – Дніпро: НГУ, 2017. – 247 с.
8. Проців В.В. Прикладна комп'ютерна графіка [Текст]: Навч. посібник / В.В. Проців, К.А. Зіборов, К.М. Бас, Г.К. Ванжа; М-во освіти і наук, Нац. гірн. ун-т. - Д.: НГУ, 2016. - 187 с.
9. Kopf, Johannes and Lischinski, Dani. Depixelizing Pixel Art (англ.) // ACM Trans. Graph.. — 2011. — Vol. 30, no. 4. — P. 99:1--99:8.
10. Giachetti, Andrea and Asuni, Nicola. Real-Time Artifact-Free Image Upscaling (англ.) // Trans. Img. Proc.. — 2011. — Vol. 20, no. 10. — P. 2760—2768.
11. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». Lecture Notes on Data Engineering and Communications Technologies, 2023, 178, pp. 208–223.

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		108

12. Smirnov, O., Neskorođieva, T., Fedorov, E., Rudakov, K., Neskorođieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» CEUR Workshop Proceedings, Volume 3187, 2022,

13. Smirnov O., Smirnova T., Anas M. Al-Oraiqat, Drieiev O., Polishchuk L., Sheroz Khan, Yassin M. Y. Hasan, Aladdein M. Amro, Hazim S. AlRawashdeh «Method for Determining Treated Metal Surface Quality Using Computer Vision Technology». Sensors (Basel, Switzerland) Volume 22, Issue 16, 6223, 2022.

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

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

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

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

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

19. Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and

cybersecurity of virtual cloud resources». Journal of theoretical and applied information technology Vol.98. No 21, 2020, P. 3334-3346.

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

21. Smirnov O., Kuznetsov A., Onikiychuk A., Makushenko T., Anisimova O., Arischenko A. «Adaptive pseudo-random sequence generation for spread spectrum image steganography». 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 161-165.

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

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

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

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

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

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

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

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

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

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

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

33. Smirnov, O., Kuznetsov, A., Kavun, S., Babenko, B., Nakisko, O., Kuznetsova, K., «Malware Correlation Monitoring in Computer Networks of Promising

Smart Grids», 2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS), Kyiv, Ukraine April 17-19, 2019 P. 347-352.

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

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

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

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

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

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

40. Смірнов О.А., Дреєва Г.М., Дреєв О.М., Смірнова Т.В. «Фрактальний аналіз генератора самоподібного трафіку на основі ланцюга Маркова». Центральнoукраїнський науковий вісник. Технічні науки. № 2(33). с. 161-172, 2019.

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		112

41. О. Смірнов, Є. Деменко, О. Онікійчук, А. Арищенко, Л. Горбачова, «Формування псевдовипадкових послідовностей для приховування даних в зображеннях» Комп'ютерні науки та кібербезпека. № 4. С. 30-37. 2019.

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

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

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

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

46. Смірнов О.А., Кавун С.В., Коваленко О.В., Дреєв О.М. Мережні інформаційні технології. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 159 с.

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

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		113

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

49. Смірнов О.А., Дреєв О.М. Порівняння бітових щільностей при використанні різних методів кодування інформації. Збірник наукових праць "Системи обробки інформації". - Випуск 2 (118). т.2. - Х.: ХУПС - 2014. - С. 64-67

50. Смірнов О.А., Дреєв О.М. Порівняння бітових щільностей при використанні різних методів кодування інформації. Збірник тез VI міжнародної науково-практичної конференції "Проблеми та перспективи розвитку ІТ-індустрії". м. Харків. 17-18 квітня 2014р. – Харків: ХНЄУ. - 2014. - С. 240.

51. Смірнов О.А., Коваленко О.В., Кожанова А.С., Лешко О.Л., Константинова Л.В. Основи системного програмування. Навчальний посібник для студентів вищих навчальних закладів напрямів підготовки 8.050102 «Комп'ютерна інженерія». За ред. Коваленка О.В., Гриф "Навчальний посібник" надано у відповідності з листом Міністерства освіти і науки України від 26.02.2013 року № 1/11-4368. – Кіровоград: КНТУ 2013. – 257с.

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

53. Смірнов О.А., Доренський О.П., Дреєв О.М. Аналіз процесів стиснення та відновлення зображень на основі цифрових методів. Наука і техніка Повітряних сил Збройних Сил України. – Випуск 3(12). – Х.: ХУПС. – 2013. – С.122-127.

					ВКРМ-122.23.0070.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		114

Додаток А
(обов'язковий)

Технічне завдання

Зміст

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

					ВКРМ-122.23.0070.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Трапезнікова Н.В.				<i>Дослідження та програмна реалізація системи передачі даних в IRC-мережах</i>	Літ.	Аркуш	Аркушів
Перевірів	Дресва Г.М.					М	1	6
Н. Контр.	Коваленко А.С.				ЦНТУ КН-22МЗ			
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи передачі даних в IRC-мережах.

2 Підстава для розробки

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

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи передачі даних в IRC-мережах.

4 Джерела розробки

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

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

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

					ВКРМ-122.23.0070.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

- техніко-економічне обґрунтування доцільності прийнятого до розробки програмного забезпечення;
- аналіз умов праці;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

- програмну реалізацію системи передачі даних в IRC-мережах;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

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

5.4 Вимоги до архітектури

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

5.5 Вимоги до надійності

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

					ВКРМ-122.23.0070.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

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

5.7 Вимоги до складу та параметрів технічних засобів

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

5.8 Вимоги до інформаційної і програмної сумісності

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

5.8.1 Обладнання

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

5.8.2 Мова програмування

Середовище Delphi 10.4 Sydney.

					ВКРМ-122.23.0070.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

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

7 Економічні вимоги

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

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

8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинен бути розглянутий аналіз санітарно-гігієнічних умов праці на робочому місці програміста.

					ВКРМ-122.23.0070.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

9 Перелік документів, що розробляються

– Наукова новизна	– 1 аркуш.
– Структурна схема системи	– 1 аркуш.
– Функціональна схема системи	– 1 аркуш.
– Діаграма процесів	– 1 аркуш.
– Блок-схема алгоритму роботи програми	– 2 аркуша.
– Показники економічної ефективності	– 1 аркуш.
– Пояснювальна записка	– 114 аркушів.

10 Етапи розробки

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

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

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

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Розрахунок з техніко-економічного обґрунтування.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

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

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 15.12.2023 р.

					ВКРМ-122.23.0070.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

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

_____ Дреєва Г.М.

*Дослідження та програмна реалізація
системи передачі даних в IRC-мережах*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 86

Літера: РП

Кропивницький – 2023 року

// IRC_netCommandsUnit.pas - модуль обработки команд

```

unit IRC_netCommandsUnit;

interface

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

type
  TIRC_netCommandsForm = class(TForm)
    IRC_net_Commands_Label: TLabel;
    IRC_net_Commands_OK_Button: TButton;
    IRC_net_Commands_Close_Button: TButton;
    IRC_net_Commands_GroupBox: TGroupBox;
    IRC_net_Commands_Edit: TEdit;
    procedure IRC_net_Commands_OK_ButtonClick(Sender: TObject);
    procedure IRC_net_Commands_Close_ButtonClick(Sender: TObject);
    procedure IRC_net_Commands_EditKeyDown(Sender: TObject; var Key: Word;
      Shift: TShiftState);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  IRC_netCommandsForm: TIRC_netCommandsForm;

implementation

uses InstrumentsUnit;

{$R *.dfm}

procedure TIRC_netCommandsForm.IRC_net_Commands_OK_ButtonClick(
  Sender: TObject);
var
  i:integer;
begin
  if (IRC_net_Commands_Edit.Text='') or (IRC_net_Commands_Edit.Text='#') then
  close
  else
  begin
    i:=Pos(' ',IRC_net_Commands_Edit.Text);
    try
      // Включити режим AWAY
      if IRC_netCommandsForm.Caption='Включити режим AWAY' then
      begin
        Form1.IdIRC_NET.WriteLine('AWAY :'+IRC_net_Commands_Edit.Text);
        away_message:=IRC_net_Commands_Edit.Text;
      end;

      // Зайти в канал
      if IRC_netCommandsForm.Caption='Зайти в канал' then
      Form1.IdIRC_NET.Join(IRC_net_Commands_Edit.Text);

      // Покинути канал
      if IRC_netCommandsForm.Caption='Покинути канал' then
      Form1.IdIRC_NET.Part(IRC_net_Commands_Edit.Text);

      // Послати приват
      if IRC_netCommandsForm.Caption='Послати приват' then
      begin

```

```

Form1.IdIRC_NET.WriteLine('PRIVMSG '+Copy(IRC_net_Commands_Edit.Text,1,
i-1)'+
:'+Copy(IRC_net_Commands_Edit.Text,i+1,Length(IRC_net_Commands_Edit.Text)-i));
    With Form1 Do
TIRC_NETMemo (FindComponent (Form1.IRC_net_PageControl.ActivePage.Name+'_IRC_NETMe
mo')).AddFormattedString (TAG_COLOR+'1<'+Copy(IRC_net_Commands_Edit.Text,1, i-
1)'+> '+Copy(IRC_net_Commands_Edit.Text,i+1,Length(IRC_net_Commands_Edit.Text)-
i));
        // Робота з файлами лога
        Form1.IRC_NETLogs (Copy(IRC_net_Commands_Edit.Text,1, i-
1),Copy(IRC_net_Commands_Edit.Text,1, i-
1),Copy(IRC_net_Commands_Edit.Text,i+1,Length(IRC_net_Commands_Edit.Text)-i));
        end;

// Послати Notice
if IRC_netCommandsForm.Caption='Послати Notice' then
begin
    Form1.IdIRC_NET.WriteLine('NOTICE '+Copy(IRC_net_Commands_Edit.Text,1,
i-1)'+
:'+Copy(IRC_net_Commands_Edit.Text,i+1,Length(IRC_net_Commands_Edit.Text)-i));
        With Form1 Do
TIRC_NETMemo (FindComponent (Form1.IRC_net_PageControl.ActivePage.Name+'_IRC_NETMe
mo')).AddFormattedString (TAG_COLOR+'7* - '+Copy(IRC_net_Commands_Edit.Text,1, i-
1)'+ - '+Copy(IRC_net_Commands_Edit.Text,i+1,Length(IRC_net_Commands_Edit.Text)-
i));
        end;

// Послати Action
if IRC_netCommandsForm.Caption='Послати Action' then
begin
    Form1.IdIRC_NET.WriteLine('PRIVMSG '+Copy(IRC_net_Commands_Edit.Text,1,
i-1)'+ :[] ACTION
'+Copy(IRC_net_Commands_Edit.Text,i+1,Length(IRC_net_Commands_Edit.Text)-i)+'[]
');
        With Form1 Do
TIRC_NETMemo (FindComponent (Form1.IRC_net_PageControl.ActivePage.Name+'_IRC_NETMe
mo')).AddFormattedString (TAG_COLOR+'6* '+Form1.IdIRC_NET.Nick+
'+Copy(IRC_net_Commands_Edit.Text,i+1,Length(IRC_net_Commands_Edit.Text)-i));
        // Робота з файлами лога
        Form1.IRC_NETLogs (Copy(IRC_net_Commands_Edit.Text,1, i-
1),Form1.IdIRC_NET.Nick,Copy(IRC_net_Commands_Edit.Text,i+1,Length(IRC_net_Comma
nds_Edit.Text)-i));
        end;

// Whois користувача (online)
if IRC_netCommandsForm.Caption='Whois користувача (online)' then
Form1.IdIRC_NET.WriteLine('WHOIS '+IRC_net_Commands_Edit.Text);

// Whowas користувача (offline)
if IRC_netCommandsForm.Caption='Whowas користувача (offline)' then
Form1.IdIRC_NET.WriteLine('HOWAS '+IRC_net_Commands_Edit.Text);

// Перемінити нік
if IRC_netCommandsForm.Caption='Перемінити нік' then
Form1.IdIRC_NET.WriteLine('NICK '+IRC_net_Commands_Edit.Text);

// Послати STCP запит Ping
if IRC_netCommandsForm.Caption='Послати STCP запит Ping' then
Form1.IdIRC_NET.WriteLine('PRIVMSG '+IRC_net_Commands_Edit.Text+' :[] PING[] ');

// Послати STCP запит Time
if IRC_netCommandsForm.Caption='Послати STCP запит Time' then
Form1.IdIRC_NET.WriteLine('PRIVMSG '+IRC_net_Commands_Edit.Text+' :[] TIME[] ');

// Послати STCP запит Version
if IRC_netCommandsForm.Caption='Послати STCP запит Version' then
Form1.IdIRC_NET.WriteLine('PRIVMSG '+IRC_net_Commands_Edit.Text+' :[] VERSION[] ');

// Послати STCP запит Finfer

```

```

    if IRC_netCommandsForm.Caption='Послати CTCP запит Finger' then
Form1.IdIRC_NET.WriteLine('PRIVMSG '+IRC_net_Commands_Edit.Text+' :[] FINGER[] ');

    // Послати CTCP запит UserInfo
    if IRC_netCommandsForm.Caption='Послати CTCP запит UserInfo' then
Form1.IdIRC_NET.WriteLine('PRIVMSG '+IRC_net_Commands_Edit.Text+' :[] USERINFO[] ');

    // Послати CTCP запит ClientInfo
    if IRC_netCommandsForm.Caption='Послати CTCP запит ClientInfo' then
Form1.IdIRC_NET.WriteLine('PRIVMSG '+IRC_net_Commands_Edit.Text+' :[] CLIENTINFO[]
');

    // Послати CTCP запит Source
    if IRC_netCommandsForm.Caption='Послати CTCP запит Source' then
Form1.IdIRC_NET.WriteLine('PRIVMSG '+IRC_net_Commands_Edit.Text+' :[] SOURCE[] ');

    // Послати CTCP запит Sound
    if IRC_netCommandsForm.Caption='Послати CTCP запит Sound' then
Form1.IdIRC_NET.WriteLine('PRIVMSG '+IRC_net_Commands_Edit.Text+' :[] SOUND[] ');

    // Одержати інформацію про канал
    if IRC_netCommandsForm.Caption='Одержати інформацію про канал' then
Form1.IdIRC_NET.WriteLine('CS INFO '+IRC_net_Commands_Edit.Text+' ALL');

    // Зареєструвати нік
    if IRC_netCommandsForm.Caption='Зареєструвати нік' then
Form1.IdIRC_NET.WriteLine('NS REGISTER '+IRC_net_Commands_Edit.Text);

    // Зареєструвати канал
    if IRC_netCommandsForm.Caption='Зареєструвати канал' then
Form1.IdIRC_NET.WriteLine('CS REGISTER '+IRC_net_Commands_Edit.Text);

    Close;
except
    // Помилка
    on e:Exception do
        begin
            Form1.Reconnect(E.Message);
            Close;
        end;
end;
end;
end;

procedure TIRC_netCommandsForm.IRC_net_Commands_Close_ButtonClick(Sender:
TObject);
begin
    Close;
end;

// Вставка спецтегів
procedure TIRC_netCommandsForm.IRC_net_Commands_EditKeyDown(Sender: TObject;
var Key: Word; Shift: TShiftState);
begin
    Form1.IRC_netColor(nil, IRC_netCommandsForm.IRC_net_Commands_Edit, Key, Shift);
end;

end.
```

```
// instruments.dpr - файл проекта
```

```
program instruments;

uses
  Forms,
  InstrumentsUnit in 'InstrumentsUnit.pas' {Form1},
  IRC_netSettingsUnit in 'IRC_netSettingsUnit.pas' {IRC_netSettingsForm},
  IRC_netCommandsUnit in 'IRC_netCommandsUnit.pas' {IRC_netCommandsForm},
  IRC_netColorUnit in 'IRC_netColorUnit.pas' {IRC_netColorForm},
  IRC_netFavoriteUnit in 'IRC_netFavoriteUnit.pas' {IRC_netFavoriteForm},
  IRC_netBotUnit in 'IRC_netBotUnit.pas' {IRC_netBotForm},
  IRC_netChannelSettingsUnit in 'IRC_netChannelSettingsUnit.pas'
{IRC_netChannelSettingsForm},
  IRC_netDCCUnit in 'IRC_netDCCUnit.pas' {IRC_netDCCForm};

{$R *.res}

begin
  Application.Initialize;
  Application.CreateForm(TForm1, Form1);
  Application.CreateForm(TIRC_netSettingsForm, IRC_netSettingsForm);
  Application.CreateForm(TIRC_netCommandsForm, IRC_netCommandsForm);
  Application.CreateForm(TIRC_netColorForm, IRC_netColorForm);
  Application.CreateForm(TIRC_netFavoriteForm, IRC_netFavoriteForm);
  Application.CreateForm(TIRC_netBotForm, IRC_netBotForm);
  Application.CreateForm(TIRC_netChannelSettingsForm,
IRC_netChannelSettingsForm);
  Application.CreateForm(TIRC_netDCCForm, IRC_netDCCForm);
  Application.Run;
end.
```

```
// InstrumentsUnit.pas - модуль IRC_NET-клиенту
```

```
unit InstrumentsUnit;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Forms, Variants, shellapi,  
CommCtrl, Menus, IdBaseComponent, IdComponent, IdTCPConnection,  
IdTCPClient, IdTelnet, Classes, ComCtrls, StdCtrls, ExtCtrls,  
Controls, ImgList,  
MMSystem, WWave32, IdIRC_NET, Graphics, IniFiles, ExtActns,  
msxmldom, XMLDoc, IdAntiFreezeBase, IdAntiFreeze,  
Spin, Buttons, XPMAN, IRC_NETMemo, MSNPopUp, WinInet, Speech, Active;
```

```
const
```

```
// IRC_NET клиент  
IRC_NET_OP=1;  
IRC_NET_VOICE=2;
```

```
type
```

```
// IRC_NET клиент  
TIRC_NETUserState=(IRC_netOp, IRC_netVoice);  
TIRC_NETUserStates=set of TIRC_NETUserState;  
  
TForm1 = class(TForm)  
ImageList: TImageList;  
Wave: TWave;  
IdIRC_NET: TIdIRC_NET;  
IRC_net_Channel_ListBox_PopupMenu: TPopupMenu;  
IRC_net_Channels_ListBox_PopupMenu_Whois: TMenuItem;  
IRC_net_Channels_ListBox_PopupMenu_Privat: TMenuItem;  
N6: TMenuItem;  
IRC_net_Channels_ListBox_PopupMenu_Control: TMenuItem;  
IRC_net_Channels_ListBox_PopupMenu_CTCP: TMenuItem;  
IRC_net_Channels_ListBox_PopupMenu_DCC: TMenuItem;  
N8: TMenuItem;  
IRC_net_Channels_ListBox_PopupMenu_CTCP_: TMenuItem;  
IRC_net_Channels_ListBox_PopupMenu_CTCP_Time: TMenuItem;  
IRC_net_Channels_ListBox_PopupMenu_CTCP_Version: TMenuItem;  
IRC_net_Channels_ListBox_PopupMenu_DCC_Send: TMenuItem;  
IRC_net_Channels_ListBox_PopupMenu_DCC_Chat: TMenuItem;  
IRC_net_Channels_ListBox_PopupMenu_Control_Op: TMenuItem;  
IRC_net_Channels_ListBox_PopupMenu_Control_DeOp: TMenuItem;  
IRC_net_Channels_ListBox_PopupMenu_Control_Voice: TMenuItem;  
IRC_net_Channels_ListBox_PopupMenu_Control_DeVoice: TMenuItem;  
IRC_net_Channels_ListBox_PopupMenu_Control_Kick: TMenuItem;  
IRC_net_Channels_ListBox_PopupMenu_AddNick: TMenuItem;  
IRC_net_Channels_ListBox_PopupMenu_CTCP_Finger: TMenuItem;  
N4: TMenuItem;  
IRC_net_Channels_ListBox_PopupMenu: TPopupMenu;  
Go_To_Channel: TMenuItem;  
IRC_net_Channels_ListBox_PopupMenu_CTCP_Userinfo: TMenuItem;  
IRC_net_Channels_ListBox_PopupMenu_CTCP_ClientInfo: TMenuItem;  
IRC_net_Channels_ListBox_PopupMenu_Action: TMenuItem;  
IRC_net_Channels_ListBox_PopupMenu_Action1: TMenuItem;  
IRC_net_Channels_ListBox_PopupMenu_Action_Ger11: TMenuItem;  
IRC_net_Channels_ListBox_PopupMenu_Action_Ger12: TMenuItem;  
IRC_net_Channels_ListBox_PopupMenu_Action4: TMenuItem;  
IRC_net_Channels_ListBox_PopupMenu_Action_Ger13: TMenuItem;  
IRC_net_Channels_ListBox_PopupMenu_Action_Ger14: TMenuItem;  
IRC_net_Channels_ListBox_PopupMenu_Action2: TMenuItem;  
IRC_net_Channels_ListBox_PopupMenu_Action3: TMenuItem;  
IRC_net_Channels_ListBox_PopupMenu_CTCP_Source: TMenuItem;  
IdAntiFreeze: TIdAntiFreeze;  
IRC_net_Channels_ListBox_PopupMenu_Control_Ban: TMenuItem;  
IRC_net_Commands_PopupMenu: TPopupMenu;  
IRC_net_Commands_PopupMenu_Join: TMenuItem;
```

```
IRC_net_Commands_PopupMenu_Part: TMenuItem;
IRC_net_Commands_PopupMenu_Msg: TMenuItem;
IRC_net_Commands_PopupMenu_Notice: TMenuItem;
IRC_net_Commands_PopupMenu_Whois: TMenuItem;
IRC_net_Commands_PopupMenu_Whowas: TMenuItem;
IRC_net_Commands_PopupMenu_Nick: TMenuItem;
IRC_net_Commands_PopupMenu_CTCP: TMenuItem;
IRC_net_Commands_PopupMenu_CTCP_: TMenuItem;
IRC_net_Commands_PopupMenu_CTCP_Time: TMenuItem;
IRC_net_Commands_PopupMenu_CTCP_Version: TMenuItem;
IRC_net_Commands_PopupMenu_CTCP_Finger: TMenuItem;
IRC_net_Commands_PopupMenu_CTCP_UserInfo: TMenuItem;
IRC_net_Commands_PopupMenu_CTCP_ClientInfo: TMenuItem;
IRC_net_Commands_PopupMenu_CTCP_Source: TMenuItem;
IRC_net_Commands_PopupMenu_CTCP_Sound: TMenuItem;
IRC_net_Commands_PopupMenu_Links: TMenuItem;
IRC_net_Commands_PopupMenu_Admin: TMenuItem;
IRC_net_Commands_PopupMenu_ChannelInfo: TMenuItem;
IRC_net_Commands_PopupMenu_Me: TMenuItem;
IRC_net_Channels_ListBox_PopupMenu_CTCP_Sound: TMenuItem;
IRC_net_Commands_PopupMenu_RegNick: TMenuItem;
IRC_net_Commands_PopupMenu_RegChannel: TMenuItem;
XPManifest: TXPManifest;
IRC_NET_ImageList: TImageList;
MSNPopUp: TMSNPopUp;
IRC_net_Favorite_PopupMenu: TPopupMenu;
IRC_net_Favorite_PopupMenu_Del: TMenuItem;
IRC_net_Favorite_PopupMenu_: TMenuItem;
IRC_net_Favorite_PopupMenu_Add: TMenuItem;
IRC_net_Commands_PopupMenu_Away: TMenuItem;
IRC_net_Commands_PopupMenu_Away_On: TMenuItem;
IRC_net_Commands_PopupMenu_Away_Off: TMenuItem;
IRC_net_List_PopupMenu: TPopupMenu;
IRC_net_List_100_PopupMenu: TMenuItem;
IRC_net_List_50_PopupMenu: TMenuItem;
IRC_net_List_All_PopupMenu: TMenuItem;
IRC_net_Channels_ListBox_PopupMenu_Action6: TMenuItem;
IRC_net_Channels_ListBox_PopupMenu_Action_Gerl5: TMenuItem;
IRC_net_Channels_ListBox_PopupMenu_Action_Gerl6: TMenuItem;
IRC_net_Channels_ListBox_PopupMenu_Action_Gerl7: TMenuItem;
IRC_net_Channels_ListBox_PopupMenu_Action_Gerl8: TMenuItem;
IRC_net_Channels_ListBox_PopupMenu_Action_Gerl9: TMenuItem;
IRC_net_Channels_ListBox_PopupMenu_Action5: TMenuItem;
IRC_net_Channels_ListBox_PopupMenu_Action_Gerl: TMenuItem;
IRC_net_Channels_ListBox_PopupMenu_Action_Gerl10: TMenuItem;
IRC_net_Channels_ListBox_PopupMenu_Action_Gerl11: TMenuItem;
IRC_net_Ping_Timer: TTimer;
IRC_net_Channels_ListBox_PopupMenu_Action7: TMenuItem;
IRC_net_Channels_ListBox_PopupMenu_Action_Gerl12: TMenuItem;
IRC_net_Channels_ListBox_PopupMenu_Control_PoluOp: TMenuItem;
IRC_net_Channels_ListBox_PopupMenu_Control_DePoluOp: TMenuItem;
Inet_Timer: TTimer;
IRC_net_PageControl: TPageControl;
IRC_net_Server: TTabSheet;
IRC_net_Server_Panel: TPanel;
IRC_net_Server_Edit: TEdit;
IRC_net_Server_IRC_NETMemo: TIRC_NETMemo;
Channels: TTabSheet;
Channels_ListBox: TListBox;
Channels_IRC_NETMemo: TIRC_NETMemo;
IRC_net_Top_Panel: TPanel;
IRC_net_Close_SpeedButton: TSpeedButton;
IRC_net_Faq_SpeedButton: TSpeedButton;
IRC_net_Settings_SpeedButton: TSpeedButton;
IRC_net_Channels_SpeedButton: TSpeedButton;
IRC_net_Commands_SpeedButton: TSpeedButton;
IRC_net_Clear_SpeedButton: TSpeedButton;
IRC_net_Favorite_SpeedButton: TSpeedButton;
IRC_net_Log_SpeedButton: TSpeedButton;
```

```

IRC_net_Bot_SpeedButton: TSpeedButton;
IRC_net_Away_Status_Label: TLabel;
IRC_net_ChannelSettings_SpeedButton: TSpeedButton;
IRC_net_Nik_GroupBox: TGroupBox;
IRC_net_Nik_Edit: TEdit;
IRC_net_Password_GroupBox: TGroupBox;
IRC_net_Password_Edit: TEdit;
IRC_net_Server_Address_GroupBox: TGroupBox;
IRC_net_Server_Address_Combobox: TComboBox;
IRC_net_Connected_Button: TButton;
StatusBar: TStatusBar;
Speak_GroupBox: TGroupBox;
Speak_Combobox: TComboBox;

procedure FormCreate(Sender: TObject);
procedure IRC_net_Connected_ButtonClick(Sender: TObject);
procedure IdIRC_NETConnected(Sender: TObject);
procedure IdIRC_NETDisconnected(Sender: TObject);
procedure IdIRC_NETJoin(Sender: TObject; AUser: TIdIRC_NETUser;
  AChannel: TIdIRC_NETChannel);
procedure IdIRC_NETPart(Sender: TObject; AUser: TIdIRC_NETUser;
  AChannel: TIdIRC_NETChannel);
procedure IdIRC_NETRaw(Sender: TObject; AUser: TIdIRC_NETUser; ACommand,
  AContent: String; var Suppress: Boolean);
procedure IRC_net_Channels_MessageText_EditKeyPress(Sender: TObject;
  var Key: Char);
procedure IdIRC_NETKicked(Sender: TObject; AUser: TIdIRC_NETUser;
  AChannel: TIdIRC_NETChannel);
procedure IdIRC_NETKick(Sender: TObject; AUser, AVictim: TIdIRC_NETUser;
  AChannel: TIdIRC_NETChannel);
procedure IdIRC_NETJoined(Sender: TObject; AChannel: TIdIRC_NETChannel);
procedure IRC_net_Close_SpeedButtonClick(Sender: TObject);
procedure IRC_net_ServerShow(Sender: TObject);
procedure IRC_net_Channel_Show(Sender: TObject);
procedure IRC_net_Privat_Show(Sender: TObject);
procedure ChannelsShow(Sender: TObject);
procedure IdIRC_NETQuit(Sender: TObject; AUser: TIdIRC_NETUser);
procedure IdIRC_NETStatus(ASender: TObject; const AStatus: TIdStatus;
  const AStatusText: String);
procedure IdIRC_NETSocketError(Sender: TObject; ASocket, AMsg: String);
procedure IdIRC_NETError(Sender: TObject; AUser: TIdIRC_NETUser; ANumeric,
  AError: String);
procedure IdIRC_NETAction(Sender: TObject; AUser: TIdIRC_NETUser;
  AChannel: TIdIRC_NETChannel; Content: String);
procedure IdIRC_NETMessage(Sender: TObject; AUser: TIdIRC_NETUser;
  AChannel: TIdIRC_NETChannel; Content: String);
procedure IdIRC_NETNames(Sender: TObject; AUsers: TIdIRC_NETUsers;
  AChannel: TIdIRC_NETChannel);
procedure IdIRC_NETPingPong(Sender: TObject);
procedure IdIRC_NETTCPQuery(Sender: TObject; User: TIdIRC_NETUser;
  AChannel: TIdIRC_NETChannel; Command, Args: String;
  var ASuppress: Boolean);
procedure IdIRC_NETTCPReply(Sender: TObject; AUser: TIdIRC_NETUser;
  AChannel: TIdIRC_NETChannel; Command, Args: String);
procedure IdIRC_NETNotice(Sender: TObject; AUser: TIdIRC_NETUser;
  AChannel: TIdIRC_NETChannel; Content: String);
procedure IdIRC_NETTopic(Sender: TObject; AUser: TIdIRC_NETUser;
  AChannel: TIdIRC_NETChannel; const AChanName, ATopic: String);
procedure IdIRC_NETNickChange(Sender: TObject; AUser: TIdIRC_NETUser;
  ANewNick: String);
procedure IdIRC_NETNickChanged(Sender: TObject; AOldNick: String);
procedure IdIRC_NETNicksInUse(Sender: TObject; var ANick: String);
procedure IRC_net_Settings_SpeedButtonClick(Sender: TObject);
procedure IRC_net_Channels_EditKeyDown(Sender: TObject;
  var Key: Word; Shift: TShiftState);
procedure IRC_net_Channels_ListBoxClick(Sender: TObject);
procedure IdIRC_NETSystem(Sender: TObject; AUser: TIdIRC_NETUser;
  ACmdCode: Integer; ACommand, AContent: String);

```

```

procedure IdIRC_NETUserMode(Sender: TObject; AModes: String);
procedure IdIRC_NETReceive(Sender: TObject; ACommand: String);
procedure IRC_net_Channels_ListBox_PopupMenu_AddNickClick(Sender: TObject);
procedure IRC_net_Channels_ListBox_PopupMenu_Control_KickClick(Sender:
TObject);
procedure IRC_net_Channels_ListBox_PopupMenu_Control_OpClick(
Sender: TObject);
procedure IRC_net_Channels_ListBox_PopupMenu_Control_DeOpClick(Sender:
TObject);
procedure IRC_net_Channels_ListBox_PopupMenu_Control_VoiceClick(Sender:
TObject);
procedure IRC_net_Channels_ListBox_PopupMenu_Control_DeVoiceClick(Sender:
TObject);
procedure IdIRC_NETUnknownCommand(Sender: TObject; AUser: TIdIRC_NETUser;
ACommand, AContent: String);
procedure IdIRC_NETSend(Sender: TObject; ACommand: String);
procedure IdIRC_NETChannelMode(Sender: TObject; AUser: TIdIRC_NETUser;
AChannel: TIdIRC_NETChannel; AChanName, AModes: String);
procedure IRC_net_Channels_ListBox_PopupMenu_WhoisClick(Sender: TObject);
procedure IRC_net_Channels_ListBox_PopupMenu_CTCP_Click(
Sender: TObject);
procedure Go_To_ChannelClick(Sender: TObject);
procedure IRC_net_Channels_ListBox_PopupMenu_PrivatClick(Sender: TObject);
procedure IRC_net_Faq_SpeedButtonClick(Sender: TObject);
procedure IdIRC_NETInvite(Sender: TObject; AUser: TIdIRC_NETUser;
AChannel: String);
procedure FormDestroy(Sender: TObject);
procedure IdIRC_NETDCCAccept(Sender: TObject; ANick, APort, AFileName,
APosition: String);
procedure IdIRC_NETDCCChat(Sender: TObject; ANick, AIP, APort: String);
procedure IdIRC_NETDCCResume(Sender: TObject; ANick, APort, AFileName,
APosition: String);
procedure IdIRC_NETDCCSend(Sender: TObject; ANick, AIP, APort, AFileName,
AFileSize: String);
procedure IRC_net_Channels_ListBox_PopupMenu_Action_Click(Sender: TObject);
procedure IRC_net_Clear_SpeedButtonClick(Sender: TObject);
procedure Channels_ListBoxClick(Sender: TObject);
procedure IRC_net_Channels_ListBox_PopupMenu_Control_BanClick(
Sender: TObject);
procedure IRC_net_Commands_SpeedButtonClick(Sender: TObject);
procedure IRC_net_Commands_PopupMenu_JoinClick(Sender: TObject);
procedure IRC_net_Commands_PopupMenu_PartClick(Sender: TObject);
procedure IRC_net_Commands_PopupMenu_MsgClick(Sender: TObject);
procedure IRC_net_Commands_PopupMenu_NoticeClick(Sender: TObject);
procedure IRC_net_Commands_PopupMenu_MeClick(Sender: TObject);
procedure IRC_net_Commands_PopupMenu_WhoisClick(Sender: TObject);
procedure IRC_net_Commands_PopupMenu_WhowasClick(Sender: TObject);
procedure IRC_net_Commands_PopupMenu_NickClick(Sender: TObject);
procedure IRC_net_Commands_PopupMenu_CTCP_Click(Sender: TObject);
procedure IRC_net_Commands_PopupMenu_LinksClick(Sender: TObject);
procedure IRC_net_Commands_PopupMenu_AdminClick(Sender: TObject);
procedure IRC_net_Commands_PopupMenu_ChannelInfoClick(Sender: TObject);
procedure IRC_net_Commands_PopupMenu_RegNickClick(Sender: TObject);
procedure IRC_net_Commands_PopupMenu_RegChannelClick(Sender: TObject);
procedure IRC_net_Favorite_SpeedButtonClick(Sender: TObject);
procedure IRC_net_Favorite_PopupMenu_AddClick(Sender: TObject);
procedure IRC_net_Favorite_PopupMenu_DelClick(Sender: TObject);
procedure IdIRC_NETUnAway(Sender: TObject; AContent: String);
procedure IdIRC_NETNowAway(Sender: TObject; AContent: String);
procedure IRC_net_Commands_PopupMenu_Away_OnClick(Sender: TObject);
procedure IRC_net_Commands_PopupMenu_Away_OffClick(Sender: TObject);
procedure IRC_netNewPrivatPage(Caption: String; UserNick: String;
MessageText: String; PageImage: Integer);
procedure IRC_net_List_100_PopupMenuClick(Sender: TObject);
procedure IRC_net_List_50_PopupMenuClick(Sender: TObject);
procedure IRC_net_List_All_PopupMenuClick(Sender: TObject);
procedure IRC_net_Channels_SpeedButtonClick(Sender: TObject);
procedure IRC_net_Ping_TimerTimer(Sender: TObject);
procedure IRC_net_Log_SpeedButtonClick(Sender: TObject);

```

```

procedure Reconnect(Err: String);
procedure IdIRC_NETParted(Sender: TObject; AChannel: TIdIRC_NETChannel);
procedure IRC_NETLogs(filename: String; User: String; Content: String);
procedure IRC_netSpeak(User: String; Text: String; Channel: Boolean);
procedure IRC_netFindText(User: String; Text: String; Channel: String);
procedure IRC_net_Channel_ListBoxMouseMove(Sender: TObject;
  Shift: TShiftState; X, Y: Integer);
procedure IRC_net_Bot_SpeedButtonClick(Sender: TObject);
procedure IRC_netBotKick(User: String; Text: String; Channel: String);
procedure IRC_netBotOp(User: String; Channel: String);
procedure IRC_netBotVoice(User: String; Channel: String);
procedure IRC_netBotCommands(User: String; Text: String; Channel: String);
procedure IRC_net_ChannelSettings_SpeedButtonClick(Sender: TObject);
procedure IRC_netColor(ComboBox: TComboBox; Edit: TEdit; var Key: Word;
Shift: TShiftState);
procedure IRC_net_Channels_ListBox_PopupMenu_Control_PoluOpClick(
  Sender: TObject);
procedure IRC_net_Channels_ListBox_PopupMenu_Control_DePoluOpClick(
  Sender: TObject);
procedure IRC_net_Server_IRC_NETMemoMouseUp(Sender: TObject;
  Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
procedure Channels_ListBoxMeasureItem(Control: TWinControl;
  Index: Integer; var Height: Integer);
procedure Channels_ListBoxDrawItem(Control: TWinControl;
  Index: Integer; Rect: TRect; State: TOwnerDrawState);
procedure Inet_TimerTimer(Sender: TObject);
procedure FormResize(Sender: TObject);
procedure Speak_ComboBoxChange(Sender: TObject);

private
  {Центральний інтерфейс, через який }
  {здійснюються всі дії з мовою}
  fITTSCentral      : ITTSCentral;
  {Інтерфейс для зв'язку з аудіоустройствами}
  fIAMM             : IAudioMultimediaDevice;
  {Інтерфейс для перебору движків}
  aTTSEnum         : ITTSEnum;
  {Показчик на параметри двіжка}
  fpModeInfo       : PTTSModeInfo;

  // Клік по меню обране в IRC_NET
  procedure ItemClick(sender:Tobject);
  // Кольоровий список каналів
  function CheckItemSize(Index: Integer): Integer;

protected

public
  { Public declarations }
end;

////////////////////////////////////
const
  // Кольоровий список каналів
  Кольори: Array [0 .. 15] TColor = ( / / MIRC_NET кольори
  $ 00FFFFFF, / / Білий
  $ 00000000 / / Чорний
  $ 007F0000, / / Синій
  $ 00009300 / / Зелений
  $ 000000ff / / Червоний
  $ 0000007F / / Коритневий
  $ 009C009C / / Бузковий
  $ 00007FFC / / Помаранчевий
  $ 0000FFFF, / / Жовтий
  $ 0000FC00 / / Світло-зелений
  $ 00939300 / / Синьо-зелений
  $ 00FFFF00 / / Блакитний

```

```
$ 00FC0000 / / Фіолетовий  
$ 00FF00FF / / Рожевий  
$ 007F7F7F / / Темно-сірий  
$ 00D2D2D2 / / світло-сірий  
);  
var  
  Form1              : TForm1;  
  // Привласнюємо початкове значення додаткової цифри до імені вкладки каналу  
  й привата  
  TabSheet_Integer  : Integer=10;  
  // Привласнюємо початкове значення кількості перепідключень  
  Reconnect_Integer : Integer=0;  
  away_message      : String;
```

implementation

```
uses  
  IRC_netSettingsUnit, IRC_netCommandsUnit, IRC_netColorUnit,  
  IRC_netFavoriteUnit,  
  IRC_netBotUnit, IRC_netChannelSettingsUnit, IRC_netDCCUnit;
```

```
{$R *.dfm}  
{$R sound.res}  
{$R IRC_net.res}
```

```
/////////////////////////////////////////////////////////////////  
// IRC_NET
```

```
// Функція видалення IRC_NET тегів  
function ParseColorString(s:string):string;
```

```
var  
  i, j, m, n, o : integer;  
begin  
  Result:='';  
  if s='' then exit;  
  i:=0;  
  while i<=Length(s) do  
  begin  
    inc(i);  
    if i>1 then  
      if ord(s[ i-1])<=31 then dec(i);
```

```
//жирний  
  if s[i]=#2 then Delete(s,i,1);  
//підкреслений  
  if s[i]=#31 then Delete(s,i,1);  
//інвернований  
  if s[i]=#22 then Delete(s,i,1);  
//рівний  
  if s[i]=#15 then Delete(s,i,1);  
//кольори  
  if s[i]=#3 then  
  begin  
    Delete(s,i,1);
```

```
    try  
      j:=StrToInt(s[i]);  
    except  
      j:=-1;  
    end;
```

```
    try  
      m:=StrToInt(s[i+1]);  
    except
```

```

m:=-1;
end;

if ((m=-1) and (s[i+1]=',')) or ((m>-1) and (s[i+2]=',')) then //if
begin

    try
    if m=-1 then
        n:=StrToInt(s[i+2])
    else
        n:=StrToInt(s[i+3]);
    except
    n:=-1;
    end;

    try
    if m=-1 then
        o:=StrToInt(s[i+3])
    else
        o:=StrToInt(s[i+4]);
    except
    o:=-1;
    end;

end //if

else

begin
n:=-1;o:=-1;
end; //if
if (j=-1) then Continue;
if ((j>-1) and (m>-1)) then Delete(s,i,2);
if ((j>-1) and (m=-1)) then Delete(s,i,1);
if ((n>-1) and (o>-1)) then Delete(s,i,3);
if ((n>-1) and (o=-1)) then Delete(s,i,2);
end;
end;

Result:=s;
end;

// Проказування повідомлень активного каналу й приватів
procedure TForm1.IRC_netSpeak(User: String; Text: String; Channel: Boolean);
var
    SData : TSDData;
    BufRich : String;
begin
    try
        if Channel=true then BufRich:=User+' говорить '+Text else BufRich:=User+'
говорить у приват '+Text;
        SData.dwSize := length(BufRich) + 1;
        SData.pData := pChar(BufRich);
        fITTSCentral.TextData(CHARSET_TEXT, 1, SData, nil, IID_ITTSBufNotifySink);
    except
        end;
end;

// Вивід повідомлення при згадуванні слів зі списку
procedure TForm1.IRC_netFindText(User: String; Text: String; Channel: String);
var
    i : Integer;
begin
if IRC_netSettingsForm.IRC_netSettings_Show_Find_Messages_CheckBox.Checked then
begin
for i:=0 to
IRC_netSettingsForm.IRC_netSettings_Show_Find_Messages_Memo.Lines.Count-1 do
begin

```

```

        while
IRC_netSettingsForm.IRC_netSettings_Show_Find_Messages_Memo.Lines[i][Length(IRC_
netSettingsForm.IRC_netSettings_Show_Find_Messages_Memo.Lines[i])]=' ' do

IRC_netSettingsForm.IRC_netSettings_Show_Find_Messages_Memo.Lines[i]:=Copy(IRC_n
etSettingsForm.IRC_netSettings_Show_Find_Messages_Memo.Lines[i],1,Length(IRC_net
SettingsForm.IRC_netSettings_Show_Find_Messages_Memo.Lines[i])-1);
        if
StrPos(PChar(AnsiUpperCase(ParseColorString(Text))),PChar(AnsiUpperCase(IRC_netS
ettingsForm.IRC_netSettings_Show_Find_Messages_Memo.Lines[i])))<>nil then
        begin
            MSNPopUp.Title:='IRC_NET '+Channel+' '+User;
            MSNPopUp.Text:=ParseColorString(Text);
            MSNPopUp.GradientColor1:=clInfoBk;
            MSNPopUp.ShowPopUp;
        end;
    end;
end;
end;

// Банити користувачів за використання слів зі списку
procedure TForm1.IRC_netBotKick(User: String; Text: String; Channel: String);
var
    i : Integer;
begin
    if IRC_netBotForm.IRC_netBot_CheckBox.Checked then
        begin
            if IRC_netBotForm.IRC_netBot_Kick_CheckBox.Checked then
                begin
                    for i:=0 to IRC_netBotForm.IRC_netBot_Kick_Memo.Lines.Count-1 do
                        begin
                            while
IRC_netBotForm.IRC_netBot_Kick_Memo.Lines[i][Length(IRC_netBotForm.IRC_netBot_Ki
ck_Memo.Lines[i])]=' ' do

IRC_netBotForm.IRC_netBot_Kick_Memo.Lines[i]:=Copy(IRC_netBotForm.IRC_netBot_Kic
k_Memo.Lines[i],1,Length(IRC_netBotForm.IRC_netBot_Kick_Memo.Lines[i])-1);
                            if
StrPos(PChar(AnsiUpperCase(ParseColorString(Text))),PChar(AnsiUpperCase(IRC_netB
otForm.IRC_netBot_Kick_Memo.Lines[i])))<>nil then
                                begin
                                    // Для всіх каналів
                                    if IRC_netBotForm.IRC_netBot_Channels_RadioButton.Checked
then
                                        begin
                                            try
                                                IdIRC_NET.WriteLine('KICK '+Channel+' '+User+'
:'+IRC_netBotForm.IRC_netBot_Kick_Edit.Text);
                                            except
                                                on e:Exception do Reconnect(E.Message);
                                            end;
                                        end;
                                    // Для одного каналу
                                    if IRC_netBotForm.IRC_netBot_Channel_RadioButton.Checked
then
                                        begin
                                            begin
                                                if
AnsiUpperCase(Channel)=AnsiUpperCase(IRC_netBotForm.IRC_netBot_Channel_Edit.Text
) then
                                                    begin
                                                        try
                                                            IdIRC_NET.WriteLine('KICK '+Channel+' '+User+'
:'+IRC_netBotForm.IRC_netBot_Kick_Edit.Text);
                                                        except
                                                            on e:Exception do Reconnect(E.Message);
                                                        end;
                                                    end;
                                                end;
                                            end;
                                        end;
                                    end;
                                end;
                            end;
                        end;
                    end;
                end;
            end;
        end;
    end;
end;

```

```

        end;
    end;
end;

// Давати +o користувачам зі списку
procedure TForm1.IRC_netBotOp(User: String; Channel: String);
var
    i : Integer;
begin
    if IRC_netBotForm.IRC_netBot_CheckBox.Checked then
        begin
            if IRC_netBotForm.IRC_netBot_Op_CheckBox.Checked then
                begin
                    for i:=0 to IRC_netBotForm.IRC_netBot_Op_Memo.Lines.Count-1 do
                        begin
                            while
                                IRC_netBotForm.IRC_netBot_Op_Memo.Lines[i][Length(IRC_netBotForm.IRC_netBot_Op_Memo.Lines[i])]=' ' do

                                IRC_netBotForm.IRC_netBot_Op_Memo.Lines[i]:=Copy(IRC_netBotForm.IRC_netBot_Op_Memo.Lines[i],1,Length(IRC_netBotForm.IRC_netBot_Op_Memo.Lines[i])-1);
                                    if
                                        StrPos(PChar(AnsiUpperCase(User)),PChar(AnsiUpperCase(IRC_netBotForm.IRC_netBot_Op_Memo.Lines[i])))<>nil then
                                            begin
                                                // Для всіх каналів
                                                if IRC_netBotForm.IRC_netBot_Channels_RadioButton.Checked
then
                                                    begin
                                                        try
                                                            IdIRC_NET.WriteLine('MODE '+Channel+' +o '+User);
                                                        except
                                                            on e:Exception do Reconnect(E.Message);
                                                        end;
                                                    end;
                                                // Для одного каналу
                                                if IRC_netBotForm.IRC_netBot_Channel_RadioButton.Checked
then
                                                    begin
                                                        if
AnsiUpperCase(Channel)=AnsiUpperCase(IRC_netBotForm.IRC_netBot_Channel_Edit.Text
) then
                                                            begin
                                                                try
                                                                    IdIRC_NET.WriteLine('MODE '+Channel+' +o '+User);
                                                                except
                                                                    on e:Exception do Reconnect(E.Message);
                                                                end;
                                                            end;
                                                        end;
                                                    end;
                                                end;
                                            end;
                                        end;
                                    end;
                                end;
                            end;
                        end;
                    end;
                end;
            end;
        end;
    end;

// Давати +v користувачам зі списку
procedure TForm1.IRC_netBotVoice(User: String; Channel: String);
var
    i : Integer;
begin
    if IRC_netBotForm.IRC_netBot_CheckBox.Checked then
        begin
            if IRC_netBotForm.IRC_netBot_Voice_CheckBox.Checked then
                begin
                    for i:=0 to IRC_netBotForm.IRC_netBot_Voice_Memo.Lines.Count-1 do
                        begin

```

```

        while
        IRC_netBotForm.IRC_netBot_Voice_Memo.Lines[i][Length(IRC_netBotForm.IRC_netBot_Voice_Memo.Lines[i])]=' ' do

        IRC_netBotForm.IRC_netBot_Voice_Memo.Lines[i]:=Copy(IRC_netBotForm.IRC_netBot_Voice_Memo.Lines[i],1,Length(IRC_netBotForm.IRC_netBot_Voice_Memo.Lines[i])-1);
        if
        StrPos(PChar(AnsiUpperCase(User)),PChar(AnsiUpperCase(IRC_netBotForm.IRC_netBot_Voice_Memo.Lines[i])))<>nil then
        begin
            // Для всіх каналів
            if IRC_netBotForm.IRC_netBot_Channels_RadioButton.Checked
        then
            begin
                try
                    IdIRC_NET.WriteLine('MODE '+Channel+' +v '+User);
                except
                    on e:Exception do Reconnect(E.Message);
                end;
            end;
            // Для одного каналу
            if IRC_netBotForm.IRC_netBot_Channel_RadioButton.Checked
        then
            begin
                if
                AnsiUpperCase(Channel)=AnsiUpperCase(IRC_netBotForm.IRC_netBot_Channel_Edit.Text) then
                begin
                    try
                        IdIRC_NET.WriteLine('MODE '+Channel+' +v '+User);
                    except
                        on e:Exception do Reconnect(E.Message);
                    end;
                end;
            end;
        end;
    end;
end;
end;
end;

// Команди для користувачів
procedure TForm1.IRC_netBotCommands(User: String; Text: String; Channel: String);
var
    i : Integer;
begin
    if IRC_netBotForm.IRC_netBot_CheckBox.Checked then
        begin
            // Кількість користувачів на каналі
            if IRC_netBotForm.IRC_netBot_Commands_1_CheckBox.Checked then
                begin
                    // Для всіх каналів
                    if IRC_netBotForm.IRC_netBot_Channels_RadioButton.Checked then
                        begin
                            if
                            StrPos(PChar(AnsiUpperCase(ParseColorString(Text))),PChar(AnsiUpperCase(IRC_netBotForm.IRC_netBot_Commands_1_Edit.Text)))<>nil then
                            begin
                                for i:=0 to IRC_net_PageControl.PageCount-1 do
                                    begin
                                        if IRC_net_PageControl.Pages[i].Caption=Channel then
                                            begin
                                                try
                                                    IdIRC_NET.WriteLine('PRIVMSG '+Channel+'
: '+TAG_BOLD+'Користувачів на каналі '+Channel+'
'+IntToStr(TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Count));

```

```

except
    on e:Exception do Reconnect(E.Message);
end;

TIRC_NETMemo (FindComponent (IRC_net_PageControl.Pages[i].Name+'_IRC_NETMemo')).AddFormattedString (TAG_COLOR+'1<'+IdIRC_NET.Nick+'> '+TAG_BOLD+'Користувачів на каналі '+Channel+' '+IntToStr (TListBox (FindComponent (IRC_net_PageControl.Pages[i].Name+'_ListBox')).Count));

// Робота з файлами лога
IRC_NETLogs (Channel,IdIRC_NET.Nick,'Користувачів на каналі '+Channel+' '+IntToStr (TListBox (FindComponent (IRC_net_PageControl.Pages[i].Name+'_ListBox')).Count));

end;
end;
end;
end;
// Для одного каналу
if IRC_netBotForm.IRC_netBot_Channel_RadioButton.Checked then
begin
    if
AnsiUpperCase (Channel)=AnsiUpperCase (IRC_netBotForm.IRC_netBot_Channel_Edit.Text) then
        begin
            if
StrPos (PChar (AnsiUpperCase (ParseColorString (Text))),PChar (AnsiUpperCase (IRC_netBotForm.IRC_netBot_Commands_1_Edit.Text)))<>nil then
                begin
                    for i:=0 to IRC_net_PageControl.PageCount-1 do
                    begin
                        if IRC_net_PageControl.Pages[i].Caption=Channel then
                            begin
                                try
                                    IdIRC_NET.WriteLine ('PRIVMSG '+Channel+' :'+TAG_BOLD+'Користувачів на каналі '+Channel+' '+IntToStr (TListBox (FindComponent (IRC_net_PageControl.Pages[i].Name+'_ListBox')).Count));
                                except
                                    on e:Exception do Reconnect (E.Message);
                                end;

TIRC_NETMemo (FindComponent (IRC_net_PageControl.Pages[i].Name+'_IRC_NETMemo')).AddFormattedString (TAG_COLOR+'1<'+IdIRC_NET.Nick+'> '+TAG_BOLD+'Користувачів на каналі '+Channel+' '+IntToStr (TListBox (FindComponent (IRC_net_PageControl.Pages[i].Name+'_ListBox')).Count));

// Робота з файлами лога
IRC_NETLogs (Channel,IdIRC_NET.Nick,'Користувачів на каналі '+Channel+' '+IntToStr (TListBox (FindComponent (IRC_net_PageControl.Pages[i].Name+'_ListBox')).Count));

end;
end;
end;
end;
end;

// Дати Voice користувачеві
if IRC_netBotForm.IRC_netBot_Commands_2_CheckBox.Checked then
begin
    // Для всіх каналів
    if IRC_netBotForm.IRC_netBot_Channels_RadioButton.Checked then
        begin
            if
StrPos (PChar (AnsiUpperCase (ParseColorString (Text))),PChar (AnsiUpperCase (IRC_netBotForm.IRC_netBot_Commands_2_Edit.Text)))<>nil then
                begin

```

```

        try
            IdIRC_NET.WriteLine('MODE '+Channel+' +v '+User);
        except
            on e:Exception do Reconnect(E.Message);
        end;
    end;
end;
// Для одного каналу
if IRC_netBotForm.IRC_netBot_Channel_RadioButton.Checked then
begin
    if
AnsiUpperCase(Channel)=AnsiUpperCase(IRC_netBotForm.IRC_netBot_Channel_Edit.Text
) then
        begin
            if
StrPos(PChar(AnsiUpperCase(ParseColorString(Text))),PChar(AnsiUpperCase(IRC_netB
otForm.IRC_netBot_Commands_2_Edit.Text)))<>nil then
                begin
                    try
                        IdIRC_NET.WriteLine('MODE '+Channel+' +v '+User);
                    except
                        on e:Exception do Reconnect(E.Message);
                    end;
                end;
            end;
        end;
end;

// Забрати Voice у користувача
if IRC_netBotForm.IRC_netBot_Commands_3_CheckBox.Checked then
begin
    // Для всіх каналів
    if IRC_netBotForm.IRC_netBot_Channels_RadioButton.Checked then
    begin
        if
StrPos(PChar(AnsiUpperCase(ParseColorString(Text))),PChar(AnsiUpperCase(IRC_netB
otForm.IRC_netBot_Commands_3_Edit.Text)))<>nil then
            begin
                try
                    IdIRC_NET.WriteLine('MODE '+Channel+' -v '+User);
                except
                    on e:Exception do Reconnect(E.Message);
                end;
            end;
        end;
    // Для одного каналу
    if IRC_netBotForm.IRC_netBot_Channel_RadioButton.Checked then
    begin
        if
AnsiUpperCase(Channel)=AnsiUpperCase(IRC_netBotForm.IRC_netBot_Channel_Edit.Text
) then
            begin
                if
StrPos(PChar(AnsiUpperCase(ParseColorString(Text))),PChar(AnsiUpperCase(IRC_netB
otForm.IRC_netBot_Commands_3_Edit.Text)))<>nil then
                    begin
                        try
                            IdIRC_NET.WriteLine('MODE '+Channel+' -v '+User);
                        except
                            on e:Exception do Reconnect(E.Message);
                        end;
                    end;
                end;
            end;
        end;
end;

// Довідка по командам
if IRC_netBotForm.IRC_netBot_Commands_4_CheckBox.Checked then
begin

```

```

// Для всіх каналів
if IRC_netBotForm.IRC_netBot_Channels_RadioButton.Checked then
begin
if
StrPos (PChar (AnsiUpperCase (ParseColorString (Text))), PChar (AnsiUpperCase (IRC_netBotForm.IRC_netBot_Commands_4_Edit.Text))) <> nil then
begin
if IRC_netBotForm.IRC_netBot_Commands_1_CheckBox.Checked then
begin
for i:=0 to IRC_net_PageControl.PageCount-1 do
begin
if IRC_net_PageControl.Pages[i].Caption=Channel then
begin
try
IdIRC_NET.WriteLine ('PRIVMSG '+Channel+'
:'+TAG_BOLD+IRC_netBotForm.IRC_netBot_Commands_1_Edit.Text+' -
'+IRC_netBotForm.IRC_netBot_Commands_1_CheckBox.Caption);
except
on e:Exception do Reconnect (E.Message);
end;

TIRC_NETMemo (FindComponent (IRC_net_PageControl.Pages[i].Name+'_IRC_NETMemo')).AddFormattedString (TAG_COLOR+'1<'+IdIRC_NET.Nick+'>
:'+TAG_BOLD+IRC_netBotForm.IRC_netBot_Commands_1_Edit.Text+' -
'+IRC_netBotForm.IRC_netBot_Commands_1_CheckBox.Caption);
// Робота з файлами лога

IRC_NETLogs (Channel, IdIRC_NET.Nick, IRC_netBotForm.IRC_netBot_Commands_1_Edit.Text+' - '+IRC_netBotForm.IRC_netBot_Commands_1_CheckBox.Caption);
IdAntiFreeze.Sleep (1000);
end;
end;
end;

if IRC_netBotForm.IRC_netBot_Commands_2_CheckBox.Checked then
begin
for i:=0 to IRC_net_PageControl.PageCount-1 do
begin
if IRC_net_PageControl.Pages[i].Caption=Channel then
begin
try
IdIRC_NET.WriteLine ('PRIVMSG '+Channel+'
:'+TAG_BOLD+IRC_netBotForm.IRC_netBot_Commands_2_Edit.Text+' -
'+IRC_netBotForm.IRC_netBot_Commands_2_CheckBox.Caption);
except
on e:Exception do Reconnect (E.Message);
end;

TIRC_NETMemo (FindComponent (IRC_net_PageControl.Pages[i].Name+'_IRC_NETMemo')).AddFormattedString (TAG_COLOR+'1<'+IdIRC_NET.Nick+'>
:'+TAG_BOLD+IRC_netBotForm.IRC_netBot_Commands_2_Edit.Text+' -
'+IRC_netBotForm.IRC_netBot_Commands_2_CheckBox.Caption);
// Робота з файлами лога

IRC_NETLogs (Channel, IdIRC_NET.Nick, IRC_netBotForm.IRC_netBot_Commands_2_Edit.Text+' - '+IRC_netBotForm.IRC_netBot_Commands_2_CheckBox.Caption);
IdAntiFreeze.Sleep (1000);
end;
end;
end;

if IRC_netBotForm.IRC_netBot_Commands_3_CheckBox.Checked then
begin
for i:=0 to IRC_net_PageControl.PageCount-1 do
begin
if IRC_net_PageControl.Pages[i].Caption=Channel then
begin
try

```



```

        try
            IdIRC_NET.WriteLine('PRIVMSG '+Channel+'
:'+TAG_BOLD+IRC_netBotForm.IRC_netBot_Commands_1_Edit.Text+' -
'+IRC_netBotForm.IRC_netBot_Commands_1_CheckBox.Caption);
        except
            on e:Exception do Reconnect(E.Message);
        end;

TIRC_NETMemo (FindComponent (IRC_net_PageControl.Pages[i].Name+'_IRC_NETMemo')).Ad
dFormattedString (TAG_COLOR+'1<'+IdIRC_NET.Nick+'>
'+TAG_BOLD+IRC_netBotForm.IRC_netBot_Commands_1_Edit.Text+' -
'+IRC_netBotForm.IRC_netBot_Commands_1_CheckBox.Caption);
        // Работа з файлами лога

IRC_NETLogs (Channel, IdIRC_NET.Nick, IRC_netBotForm.IRC_netBot_Commands_1_Edit.Tex
t+' - '+IRC_netBotForm.IRC_netBot_Commands_1_CheckBox.Caption);
        IdAntiFreeze.Sleep(1000);
    end;
end;
end;

        if IRC_netBotForm.IRC_netBot_Commands_2_CheckBox.Checked
then
    begin
        for i:=0 to IRC_net_PageControl.PageCount-1 do
            begin
                if IRC_net_PageControl.Pages[i].Caption=Channel
then
                    begin
                        try
                            IdIRC_NET.WriteLine('PRIVMSG '+Channel+'
:'+TAG_BOLD+IRC_netBotForm.IRC_netBot_Commands_2_Edit.Text+' -
'+IRC_netBotForm.IRC_netBot_Commands_2_CheckBox.Caption);
                        except
                            on e:Exception do Reconnect(E.Message);
                        end;

TIRC_NETMemo (FindComponent (IRC_net_PageControl.Pages[i].Name+'_IRC_NETMemo')).Ad
dFormattedString (TAG_COLOR+'1<'+IdIRC_NET.Nick+'>
'+TAG_BOLD+IRC_netBotForm.IRC_netBot_Commands_2_Edit.Text+' -
'+IRC_netBotForm.IRC_netBot_Commands_2_CheckBox.Caption);
                    // Работа з файлами лога

IRC_NETLogs (Channel, IdIRC_NET.Nick, IRC_netBotForm.IRC_netBot_Commands_2_Edit.Tex
t+' - '+IRC_netBotForm.IRC_netBot_Commands_2_CheckBox.Caption);
                    IdAntiFreeze.Sleep(1000);
                end;
            end;
        end;

        if IRC_netBotForm.IRC_netBot_Commands_3_CheckBox.Checked
then
    begin
        for i:=0 to IRC_net_PageControl.PageCount-1 do
            begin
                if IRC_net_PageControl.Pages[i].Caption=Channel
then
                    begin
                        try
                            IdIRC_NET.WriteLine('PRIVMSG '+Channel+'
:'+TAG_BOLD+IRC_netBotForm.IRC_netBot_Commands_3_Edit.Text+' -
'+IRC_netBotForm.IRC_netBot_Commands_3_CheckBox.Caption);
                        except
                            on e:Exception do Reconnect(E.Message);
                        end;

TIRC_NETMemo (FindComponent (IRC_net_PageControl.Pages[i].Name+'_IRC_NETMemo')).Ad
dFormattedString (TAG_COLOR+'1<'+IdIRC_NET.Nick+'>

```

```

'+TAG_BOLD+IRC_netBotForm.IRC_netBot_Commands_3_Edit.Text+' -
'+IRC_netBotForm.IRC_netBot_Commands_3_CheckBox.Caption);
        // Робота з файлами лога

IRC_NETLogs(Channel,IdIRC_NET.Nick,IRC_netBotForm.IRC_netBot_Commands_3_Edit.Tex
t+' - '+IRC_netBotForm.IRC_netBot_Commands_3_CheckBox.Caption);
        IdAntiFreeze.Sleep(1000);
        end;
    end;
end;

    if IRC_netBotForm.IRC_netBot_Commands_4_CheckBox.Checked
then
    begin
        for i:=0 to IRC_net_PageControl.PageCount-1 do
        begin
            if IRC_net_PageControl.Pages[i].Caption=Channel
then
                begin
                    try
                        IdIRC_NET.WriteLine('PRIVMSG '+Channel+'
:'+TAG_BOLD+IRC_netBotForm.IRC_netBot_Commands_4_Edit.Text+' -
'+IRC_netBotForm.IRC_netBot_Commands_4_CheckBox.Caption);
                    except
                        on e:Exception do Reconnect(E.Message);
                    end;

TIRC_NETMemo(FindComponent(IRC_net_PageControl.Pages[i].Name+'_IRC_NETMemo')).Ad
dFormattedString(TAG_COLOR+'1<'+IdIRC_NET.Nick+'>
'+TAG_BOLD+IRC_netBotForm.IRC_netBot_Commands_4_Edit.Text+' -
'+IRC_netBotForm.IRC_netBot_Commands_4_CheckBox.Caption);
                // Робота з файлами лога

IRC_NETLogs(Channel,IdIRC_NET.Nick,IRC_netBotForm.IRC_netBot_Commands_4_Edit.Tex
t+' - '+IRC_netBotForm.IRC_netBot_Commands_4_CheckBox.Caption);
                IdAntiFreeze.Sleep(1000);
                end;
            end;
        end;
    end;
end;
end;
end;
end;
end;
end;
end;

// Перепідключення при помилці
//Перевірка наявності підключення до інтернету
function IsConnectedToInternet: Boolean;
var
    dwConnectionTypes: DWORD;
begin
    dwConnectionTypes:=INTERNET_CONNECTION_MODEM+INTERNET_CONNECTION_LAN+INTERNET_CO
NNECTION_PROXY;
    Result:=InternetGetConnectedState(@dwConnectionTypes,0);
end;

// Перевірка наявності Інтернету
procedure TForm1.Inet_TimerTimer(Sender: TObject);
begin
    if IsConnectedToInternet=false then
    begin
        if IdIRC_NET.Connected then
        begin
            IRC_net_Ping_Timer.Enabled:=false;

```

```

TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString (TAG_BOLD+TAG_COLOR+'4Немає підключення до Інтернету');
    IdIRC_NET.Disconnect;
end;
end;
end;

procedure TForm1.Reconnect (Err: String);
begin
    if Reconnect_Integer=3 then
        begin
            TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
            AddFormattedString (TAG_BOLD+TAG_COLOR+'4'+Err);
                IdIRC_NET.Disconnect;
                exit;
            end
        else
            begin
                TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
                AddFormattedString (TAG_BOLD+TAG_COLOR+'4'+Err);
                    Reconnect_Integer:=Reconnect_Integer+1;
                    IdIRC_NET.Disconnect;
                    IdAntiFreeze.Sleep (1000);
                    IRC_net_Connected_ButtonClick (nil);
                end;
            end;
end;

// Ведення логів каналів і приватів
procedure TForm1.IRC_NETLogs (filename: String; User: String; Content: String);
var
    LogFile      : TextFile;
begin
    if (IRC_netSettingsForm.IRC_netSettings_Servis_Logs_CheckBox.Checked) and
    (IRC_net_PageControl.TabIndex>0) then
        begin
            AssignFile (LogFile, ExtractFileDir (ParamStr (0))+'\IRC_NET\Logos\'+IdIRC_NET.Host+'
            \'+filename+'.txt');
                if
                    FileExists (ExtractFilePath (paramstr (0))+'\IRC_NET\Logos\'+IdIRC_NET.Host+'\'+file
                    name+'.txt')=false then
                        begin
                            CreateDirectory (PChar (ExtractFileDir (ParamStr (0))+'\IRC_NET'), nil);

                            CreateDirectory (PChar (ExtractFileDir (ParamStr (0))+'\IRC_NET\Logos'), nil);

                            CreateDirectory (PChar (ExtractFileDir (ParamStr (0))+'\IRC_NET\Logos\'+IdIRC_NET.Hos
                            t), nil);
                                Rewrite (LogFile);
                                CloseFile (LogFile);
                            end;
                                Append (LogFile);
                                Writeln (LogFile, '['+DateTimeToStr (Now) +']'+ ' <'+User+'>
                                '+ParseColorString (Content));
                                CloseFile (LogFile);
                            end;
                        end;
                    end;

// Процедура вставки спецтегів у поля введення
procedure TForm1.IRC_netColor (ComboBox: TComboBox; Edit: TEdit; var Key: Word;
Shift: TShiftState);
var
    C : Char;
begin
    C := UpCase (Char (Key));
    if (C in ['B', 'U', 'R', 'K', #13]) and (ssCtrl in Shift) then

```

```

begin
  case C of
    'B' : begin
      if ComboBox<>nil then ComboBox.SelText:=TAG_BOLD;
      if Edit<>nil then Edit.SelText:=TAG_BOLD;
      end;
    'U' : begin
      if ComboBox<>nil then ComboBox.SelText:=TAG_UNDERLINE;
      if Edit<>nil then Edit.SelText:=TAG_UNDERLINE;
      end;
    'R' : begin
      if ComboBox<>nil then ComboBox.SelText:=TAG_INVERSE;
      if Edit<>nil then Edit.SelText:=TAG_INVERSE;
      end;
    'K' : begin
      if ComboBox<>nil then ComboBox.SelText:=TAG_COLOR;
      if Edit<>nil then Edit.SelText:=TAG_COLOR;
      end;
  end;
  Key := 0;
end;
end;

// Процедура створення нової вкладки для привата
procedure TForm1.IRC_netNewPrivatPage(Caption: String; UserNick: String;
MessageText: String; PageImage: Integer);
var
  NewPage      : TTabSheet;
  NewPanel     : TPanel;
  NewEdit      : TEdit;
  NewIRC_NETMemo : TIRC_NETMemo;
begin
  if Caption='' then exit;
  if Caption[1]='я' then Caption[1]:='Я';

  // Створюємо вкладку для привату
  NewPage:=TTabSheet.Create(Self);
  NewPage.PageControl:=IRC_net_PageControl; // Вставляємо закладку
  NewPage.Name:='TabSheet'+IntToStr(TabSheet_Integer);
  NewPage.Caption:=Caption;
  NewPage.ImageIndex:=PageImage;
  // Процедура показу вкладки привату
  NewPage.OnShow:=IRC_net_Privat_Show;

  // Вставляємо панель
  NewPanel:=TPanel.Create(Self);
  NewPanel.Parent:=NewPage;
  NewPanel.Name:=NewPage.Name+'_Panel';
  NewPanel.BevelOuter:=bvNone;
  NewPanel.Caption:='';
  NewPanel.Height:=21;
  NewPanel.Align:=alBottom;

  NewEdit:=TEdit.Create(Self); // Вставляємо Edit
  NewEdit.Parent:=NewPanel;
  NewEdit.Name:=NewPage.Name+'_Edit';
  NewEdit.Top:=0;
  NewEdit.Width:=IRC_net_Server_Edit.Width;
  NewEdit.Left:=0;
  NewEdit.Text:='';
  NewEdit.MaxLength:=510;
  // Процедура відправлення повідомлень у приват
  NewEdit.OnKeyPress:=IRC_net_Channels_MessageText_EditKeyPress;
  // Процедура вставки спецтегів
  NewEdit.OnKeyDown:=IRC_net_Channels_EditKeyDown;

  NewIRC_NETMemo:=TIRC_NETMemo.Create(Self); //вставляємо IRC_NETMemo
  NewIRC_NETMemo.Parent:=NewPage;

```

```

NewIRC_NETMemo.Name:=NewPage.Name+'_IRC_NETMemo';
NewIRC_NETMemo.Align:=alClient;
NewIRC_NETMemo.ReadOnly:=true;
NewIRC_NETMemo.Clear;
// Перехід на Edit при відпусканні миші
NewIRC_NETMemo.OnMouseUp:=IRC_net_Server_IRC_NETMemoMouseUp;
// Смайли
NewIRC_NETMemo.SmilesInfo[0].Smile:=' :)';
NewIRC_NETMemo.SmilesInfo[0].ImageIndex:=0;
NewIRC_NETMemo.SmilesInfo[1].Smile:=' =)';
NewIRC_NETMemo.SmilesInfo[1].ImageIndex:=0;
NewIRC_NETMemo.SmilesInfo[2].Smile:=' :-)';
NewIRC_NETMemo.SmilesInfo[2].ImageIndex:=0;

NewIRC_NETMemo.SmilesInfo[3].Smile:=' (:';
NewIRC_NETMemo.SmilesInfo[3].ImageIndex:=1;
NewIRC_NETMemo.SmilesInfo[4].Smile:=' :-(';
NewIRC_NETMemo.SmilesInfo[4].ImageIndex:=1;

NewIRC_NETMemo.SmilesInfo[5].Smile:=' :D';
NewIRC_NETMemo.SmilesInfo[5].ImageIndex:=2;
NewIRC_NETMemo.SmilesInfo[6].Smile:=' :-D';
NewIRC_NETMemo.SmilesInfo[6].ImageIndex:=2;

NewIRC_NETMemo.SmilesInfo[7].Smile:=' :P';
NewIRC_NETMemo.SmilesInfo[7].ImageIndex:=3;
NewIRC_NETMemo.SmilesInfo[8].Smile:=' :-P';
NewIRC_NETMemo.SmilesInfo[8].ImageIndex:=3;

NewIRC_NETMemo.SmilesInfo[9].Smile:=' ;)';
NewIRC_NETMemo.SmilesInfo[9].ImageIndex:=4;
NewIRC_NETMemo.SmilesInfo[10].Smile:=' ;-)';
NewIRC_NETMemo.SmilesInfo[10].ImageIndex:=4;

NewIRC_NETMemo.SmilesInfo[11].Smile:=' :8)';
NewIRC_NETMemo.SmilesInfo[11].ImageIndex:=5;
NewIRC_NETMemo.SmilesInfo[12].Smile:=' : 8-8-)';
NewIRC_NETMemo.SmilesInfo[12].ImageIndex:=5;

NewIRC_NETMemo.SmilesInfo[13].Smile:=' : \';
NewIRC_NETMemo.SmilesInfo[13].ImageIndex:=6;
NewIRC_NETMemo.SmilesInfo[14].Smile:=' :- \';
NewIRC_NETMemo.SmilesInfo[14].ImageIndex:=6;

NewIRC_NETMemo.SmilesInfo[15].Smile:=' :#39+' (';
NewIRC_NETMemo.SmilesInfo[15].ImageIndex:=7;
NewIRC_NETMemo.SmilesInfo[16].Smile:=' :'+#39+' (';
NewIRC_NETMemo.SmilesInfo[16].ImageIndex:=7;

NewIRC_NETMemo.Smiles:=IRC_NET_ImageList;

// Збільшуємо значення TabSheet_Integer на 1
TabSheet_Integer:=TabSheet_Integer+1;

// Додаємо текст повідомлення
if (UserNick='') or (MessageText='') then
  exit
else NewIRC_NETMemo.AddFormattedString(TAG_COLOR+'1<'+UserNick+'>
'+MessageText);
end;

// Підключення
procedure TForm1.IRC_net_Connected_ButtonClick(Sender: TObject);
begin
  // Очищаємо стан режиму Away
  IRC_net_Away_Status_Label.Visible:=false;
  if IdIRC_NET.Connected then
  begin
    try

```

```

    IdIRC_NET.Quit(IRC_netSettingsForm.IRC_netSettings_Quit_Edit.Text+' -
'+IdIRC_NET.Replies.ClientInfo);
    except
        // Помилка
        on e:Exception do
TIRC_NETMemo (FindComponent(IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString(TAG_COLOR+'4'+E.Message);
        end;
        IdIRC_NET.Disconnect;
    end
    else
    begin
    IRC_net_Server_IRC_NETMemo.Clear;
    if (IRC_netSettingsForm.IRC_netSettings_Username_Edit.Text='') or
(IRC_netSettingsForm.IRC_netSettings_AltNick_Edit.Text='') then
    begin
    IRC_netSettingsForm.ShowModal;
    exit;
    end
    else
    if IRC_net_Server_Address_Combobox.Text='' then
    begin
    TIRC_NETMemo (FindComponent(IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString(TAG_COLOR+'4Виберіть будь ласка сервер');
    ActiveControl:=IRC_net_Server_Address_Combobox;
    exit;
    end
    else
    if IRC_net_Nik_Edit.Text='' then
    begin
    TIRC_NETMemo (FindComponent(IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString(TAG_COLOR+'4Введіть будь ласка нік');
    ActiveControl:=IRC_net_Nik_Edit;
    exit;
    end
    else
    if IRC_net_Password_Edit.Text='' then
    begin
    TIRC_NETMemo (FindComponent(IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString(TAG_COLOR+'4Введіть будь ласка пароль');
    ActiveControl:=IRC_net_Password_Edit;
    exit;
    end
    else
    IRC_net_PageControl.ActivePageIndex:=0;
    IdIRC_NET.Replies.ClientInfo:=TAG_BOLD+Form1.Caption;
    IdIRC_NET.Replies.Version:=IdIRC_NET.Replies.ClientInfo;
    IdIRC_NET.Replies.Finger:=IRC_netSettingsForm.IRC_netSettings_CTCP_Finger_Edit.T
ext;
    IdIRC_NET.Replies.UserInfo:=IRC_netSettingsForm.IRC_netSettings_CTCP_Userinfo_Ed
it.Text;
    IdIRC_NET.Nick:=IRC_net_Nik_Edit.Text;
    IdIRC_NET.AltNick:=IRC_netSettingsForm.IRC_netSettings_AltNick_Edit.Text;
    IdIRC_NET.Username:=IRC_netSettingsForm.IRC_netSettings_Username_Edit.Text;
    IdIRC_NET.RealName:=IRC_netSettingsForm.IRC_netSettings_Username_Edit.Text;
    IdIRC_NET.Password:=IRC_net_Password_Edit.Text;
    IdIRC_NET.Host:=IRC_net_Server_Address_Combobox.Text;
    IdIRC_NET.Port:=IRC_netSettingsForm.IRC_netSettings_Port_SpinEdit.Value;
    try
        IdIRC_NET.Connect(500);
    except
        on e:Exception do Reconnect(E.Message);
    end;
    end;
    end;

// Підключений
procedure TForm1.IdIRC_NETConnected(Sender: TObject);
begin

```

```

IRC_net_Connected_Button.Caption:='Відключитися';
IRC_net_Nik_Edit.Enabled:=false;
IRC_net_Password_Edit.Enabled:=false;
IRC_net_Server_Address_Combobox.Enabled:=false;
IRC_net_Server.Caption:=IdIRC_NET.Host;
Reconnect_Integer:=0;
end;

// Відключення
procedure TForm1.IdIRC_NETDisconnected(Sender: TObject);
var
  i : Integer;
begin
  IRC_net_Connected_Button.Caption:='Підключитися';
  IRC_net_Nik_Edit.Enabled:=true;
  IRC_net_Password_Edit.Enabled:=true;
  IRC_net_Server_Address_Combobox.Enabled:=true;
  // Закриття вкладок каналів, якщо не стоїть галочка про перезахід
  if IRC_netSettingsForm.IRC_netSettings_Servis_Quit_CheckBox.Checked=false then
    begin
      IRC_net_PageControl.ActivePageIndex:=0;
      for i:=1 to IRC_net_PageControl.PageCount-1 do
        begin
          if IRC_net_PageControl.PageCount-1>1 then
            IRC_net_PageControl.Pages[IRC_net_PageControl.PageCount-1].Free
          else IRC_net_PageControl.Pages[IRC_net_PageControl.PageCount-1].TabVisible:=false;
        end;
      end;
    end;
end;

// Заходимо в канал (/join) і створюємо нову вкладку з назвою каналу
procedure TForm1.IdIRC_NETJoined(Sender: TObject; AChannel: TIdIRC_NETChannel);
var
  NewPage      : TTabSheet;
  NewPanel     : TPanel;
  NewEdit      : TEdit;
  NewListBox   : TListBox;
  NewSplitter  : TSplitter;
  NewIRC_NETMemo : TIRC_NETMemo;
  i            : Integer;
begin
  // Якщо вкладка з такою назвою вже існує, то показати її, очищаємо всі поля й
  зупиняємо процедуру
  for i:=0 to IRC_net_PageControl.PageCount-1 do
    begin
      if
        AnsiUpperCase(IRC_net_PageControl.Pages[i].Caption)=AnsiUpperCase(AChannel.Name)
      then
        begin
          IRC_net_PageControl.Pages[i].TabVisible:=true;
          IRC_net_PageControl.ActivePage:=IRC_net_PageControl.Pages[i];
          Exit;
        end;
    end;
end;

// Створюємо вкладку для каналу
NewPage:=TTabSheet.Create(Self);
NewPage.PageControl:=IRC_net_PageControl; // Вставляємо закладку
NewPage.Name:='TabSheet'+IntToStr(TabSheet_Integer);
NewPage.Caption:=AChannel.Name;
NewPage.ImageIndex:=4;
// Процедура показу вкладки каналу
NewPage.OnShow:=IRC_net_Channel_Show;

// Вставляємо панель
NewPanel:=TPanel.Create(Self);
NewPanel.Parent:=NewPage;
NewPanel.Name:=NewPage.Name+'_Panel';

```

```

NewPanel.BevelOuter:=bvNone;
NewPanel.Caption:='';
NewPanel.Height:=21;
NewPanel.Align:=alBottom;

NewEdit:=TEdit.Create(Self); // Вставляємо Edit
NewEdit.Parent:=NewPanel;
NewEdit.Name:=NewPage.Name+'_Edit';
NewEdit.Top:=0;
NewEdit.Width:=IRC_net_Server_Edit.Width;
NewEdit.Left:=0;
NewEdit.Text:='';
NewEdit.MaxLength:=510;
// Процедура відправлення повідомлень на канал
NewEdit.OnKeyPress:=IRC_net_Channels_MessageText_EditKeyPress;
// Процедура вставки спецтегів
NewEdit.OnKeyDown:=IRC_net_Channels_EditKeyDown;

NewListBox:=TListBox.Create(Self); // Вставляємо ListBox
NewListBox.Parent:=NewPage;
NewListBox.Name:=NewPage.Name+'_ListBox';
NewListBox.Width:=120;
NewListBox.Align:=alRight;
NewListBox.Sorted:=true;
NewListBox.ShowHint:=true;
NewListBox.PopupMenu:=IRC_net_Channel_ListBox_PopupMenu;
// Процедура кліку за списком користувачів на каналі
NewListBox.OnClick:=IRC_net_Channels_ListBoxClick;
// Процедура подвійного кліку за списком користувачів
NewListBox.OnDblClick:=IRC_net_Channels_ListBox_PopupMenu_PrivatClick;
// Процедура показу кількості користувачів на каналі
NewListBox.OnMouseMove:=IRC_net_Channel_ListBoxMouseMove;
// Перехід на Edit при відпусканні миші
NewListBox.OnMouseUp:=IRC_net_Server_IRC_NETMemoMouseUp;

// Вставляємо Splitter
NewSplitter:=TSplitter.Create(Self); // Вставляємо Splitter
NewSplitter.Parent:=NewPage;
NewSplitter.Name:=NewPage.Name+'_Splitter';
NewSplitter.ShowHint:=true;
NewSplitter.Align:=alRight;
NewSplitter.Beveled:=true;

NewIRC_NETMemo:=TIRC_NETMemo.Create(Self); //вставляємо IRC_NETMemo
NewIRC_NETMemo.Parent:=NewPage;
NewIRC_NETMemo.Name:=NewPage.Name+'_IRC_NETMemo';
NewIRC_NETMemo.Align:=alClient;
NewIRC_NETMemo.ReadOnly:=true;
NewIRC_NETMemo.Clear;
// Перехід на Edit при відпусканні миші
NewIRC_NETMemo.OnMouseUp:=IRC_net_Server_IRC_NETMemoMouseUp;
// Процедура показу властивостей каналу
NewIRC_NETMemo.OnDblClick:=IRC_net_ChannelSettings_SpeedButtonClick;

// Смайли
NewIRC_NETMemo.SmilesInfo[0].Smile:=' :) ';
NewIRC_NETMemo.SmilesInfo[0].ImageIndex:=0;
NewIRC_NETMemo.SmilesInfo[1].Smile:=' =) ';
NewIRC_NETMemo.SmilesInfo[1].ImageIndex:=0;
NewIRC_NETMemo.SmilesInfo[2].Smile:=' :-) ';
NewIRC_NETMemo.SmilesInfo[2].ImageIndex:=0;

NewIRC_NETMemo.SmilesInfo[3].Smile:=' (: ';
NewIRC_NETMemo.SmilesInfo[3].ImageIndex:=1;
NewIRC_NETMemo.SmilesInfo[4].Smile:=' :- ( ';
NewIRC_NETMemo.SmilesInfo[4].ImageIndex:=1;

NewIRC_NETMemo.SmilesInfo[5].Smile:=' :D ';

```

```

NewIRC_NETMemo.SmilesInfo[5].ImageIndex:=2;
NewIRC_NETMemo.SmilesInfo[6].Smile:='-D';
NewIRC_NETMemo.SmilesInfo[6].ImageIndex:=2;

NewIRC_NETMemo.SmilesInfo[7].Smile:='P';
NewIRC_NETMemo.SmilesInfo[7].ImageIndex:=3;
NewIRC_NETMemo.SmilesInfo[8].Smile:='-P';
NewIRC_NETMemo.SmilesInfo[8].ImageIndex:=3;

NewIRC_NETMemo.SmilesInfo[9].Smile:=')';
NewIRC_NETMemo.SmilesInfo[9].ImageIndex:=4;
NewIRC_NETMemo.SmilesInfo[10].Smile:=';-)';
NewIRC_NETMemo.SmilesInfo[10].ImageIndex:=4;

NewIRC_NETMemo.SmilesInfo[11].Smile:='8)';
NewIRC_NETMemo.SmilesInfo[11].ImageIndex:=5;
NewIRC_NETMemo.SmilesInfo[12].Smile:=' 8-8-)';
NewIRC_NETMemo.SmilesInfo[12].ImageIndex:=5;

NewIRC_NETMemo.SmilesInfo[13].Smile:=':\';
NewIRC_NETMemo.SmilesInfo[13].ImageIndex:=6;
NewIRC_NETMemo.SmilesInfo[14].Smile:='-\'';
NewIRC_NETMemo.SmilesInfo[14].ImageIndex:=6;

NewIRC_NETMemo.SmilesInfo[15].Smile:='#39+'(';
NewIRC_NETMemo.SmilesInfo[15].ImageIndex:=7;
NewIRC_NETMemo.SmilesInfo[16].Smile:=''+#39+'(';
NewIRC_NETMemo.SmilesInfo[16].ImageIndex:=7;

NewIRC_NETMemo.Smiles:=IRC_NET_ImageList;

// Перемикаємося на нову вкладку
IRC_net_PageControl.ActivePage:=NewPage;

// Збільшуємо значення TabSheet_Integer на 1
TabSheet_Integer:=TabSheet_Integer+1;

// Робота з файлами лога
IRC_NETLogs(AChannel.Name, '', 'Захід на канал');
end;

// Одержування списку користувачів в каналі
function SneakyGetUserState(Data: Integer): TIRC_NETUserStates;
begin
  Result := [];
  if (Data and IRC_NET_OP) <> 0 then Include(Result, IRC_netOp);
  if (Data and IRC_NET_VOICE) <> 0 then Include(Result, IRC_netVoice);
end;

procedure TForm1.IdIRC_NETNames(Sender: TObject; AUsers: TIdIRC_NETUsers;
  AChannel: TIdIRC_NETChannel);
var
  iUser, iIndex : Integer;
  pState : TIRC_NETUserStates;
  i : Integer;
begin
  try
  for i:=0 to IRC_net_PageControl.PageCount-1 do
    begin
      if
      AnsiUpperCase(IRC_net_PageControl.Pages[i].Caption)=AnsiUpperCase(AChannel.Name)
      then
        begin
          TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Clear;
          for iUser := 0 to AUsers.Count - 1 do
            begin
              Application.ProcessMessages;
            end;
          end;
        end;
      end;
    end;
  end;
end;

```

```

        if AChannel.Find(AUsers.Items[iUser].Nick, iIndex) then
            begin
                pState :=
                SneakyGetUserState(Integer(AChannel.Names.Objects[iIndex]));

                if IRC_netOp in pState then
                    TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Add(
                    '@'+Ausers.Items[iUser].Nick);

                if IRC_netVoice in pState then
                    TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Add(
                    '@'+Ausers.Items[iUser].Nick);

                if pState=[] then
                    TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Add(
                    Ausers.Items[iUser].Nick);
                end;
            end;
        end;
    except
    end;
end;

// Прийом даних із сервера
// Одержуємо список каналів
// Відображаємо другий нік
// Обробка цифрових відповідей, які створюються сервером як відповідна реакція на
введені команди
procedure TForm1.IdIRC_NETRaw(Sender: TObject; AUser: TIdIRC_NETUser; ACommand,
    AContent: String; var Suppress: Boolean);
var
    b,i,n,m      : integer;
    s,d,f,g,l    : String;
begin
    for i:=0 to IRC_net_PageControl.PageCount-1 do
        begin
            while
                TIRC_NETMemo(FindComponent(IRC_net_PageControl.Pages[i].Name+'_IRC_NETMemo')).Li
                nes.Count>IRC_netSettingsForm.IRC_netSettings_Servis_MaxLines_SpinEdit.Value do
                TIRC_NETMemo(FindComponent(IRC_net_PageControl.Pages[i].Name+'_IRC_NETMemo')).Li
                nes.Delete(0);
            end;
            i:=Pos(' ',AContent);

            // Одержуємо список каналів (/list)
            // Показуємо вкладку й очищуємо список каналів
            if ACommand=IntToStr(321) then
                begin
                    if Channels.TabVisible=false then
                        begin
                            Channels.TabVisible:=true;
                            Channels_IRC_NETMemo.Clear;
                            IRC_net_PageControl.ActivePageIndex:=1;
                        end;
                    Channels_ListBox.Clear;
                end;
            // Одержуємо список каналів (/list)
            if ACommand=IntToStr(322) then
                begin
                    Application.ProcessMessages;
                    Channels_ListBox.Items.Add(Copy(AContent,i+1,Length(AContent)-i));
                end;

            // Відображаємо другий нік, якщо перший уже зайнятий
            if ACommand=IntToStr(433) then IRC_net_Nik_Edit.Text:=IdIRC_NET.AltNick;

```



```

// Ці команди вже оброблені /LIST
//if ACommand=IntToStr(321) then
TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString (Copy (AContent, i+1, Length (AContent) -i));
//if ACommand=IntToStr(322) then
TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString (Copy (AContent, i+1, Length (AContent) -i));
//if ACommand=IntToStr(323) then
TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString (Copy (AContent, i+1, Length (AContent) -i));

// Ці команди вже оброблені в процедурі IdIRC_NETChannelMode
// Одержання режимів каналу
if (ACommand=IntToStr(324)) and (IRC_netChannelSettingsForm.Showing) then
begin
    s:=Copy (AContent, i+1, Length (AContent) -i);
    n:=Pos (' ', s);
    d:=Copy (s, n+1, Length (s) -n);
    m:=Pos (' ', d);
    f:=Copy (d, 0, m-1);
    g:=Copy (d, m+1, Length (d) -m);

    // +t
    if StrPos (PChar (AnsiUpperCase (f)), PChar (AnsiUpperCase ('t')))<>nil then
IRC_netChannelSettingsForm.IRC_netChannelSettings_Mode_t_CheckBox.Checked:=true;
    // +n
    if StrPos (PChar (AnsiUpperCase (f)), PChar (AnsiUpperCase ('n')))<>nil then
IRC_netChannelSettingsForm.IRC_netChannelSettings_Mode_n_CheckBox.Checked:=true;
    // +i
    if StrPos (PChar (AnsiUpperCase (f)), PChar (AnsiUpperCase ('i')))<>nil then
IRC_netChannelSettingsForm.IRC_netChannelSettings_Mode_i_CheckBox.Checked:=true;
    // +m
    if StrPos (PChar (AnsiUpperCase (f)), PChar (AnsiUpperCase ('m')))<>nil then
IRC_netChannelSettingsForm.IRC_netChannelSettings_Mode_m_CheckBox.Checked:=true;
    // +p
    if StrPos (PChar (AnsiUpperCase (f)), PChar (AnsiUpperCase ('p')))<>nil then
IRC_netChannelSettingsForm.IRC_netChannelSettings_Mode_p_CheckBox.Checked:=true;
    // +s
    if StrPos (PChar (AnsiUpperCase (f)), PChar (AnsiUpperCase ('s')))<>nil then
IRC_netChannelSettingsForm.IRC_netChannelSettings_Mode_s_CheckBox.Checked:=true;
    // +l
    if StrPos (PChar (AnsiUpperCase (f)), PChar (AnsiUpperCase ('l')))<>nil then
begin
IRC_netChannelSettingsForm.IRC_netChannelSettings_Mode_l_CheckBox.Checked:=true;
    if StrPos (PChar (AnsiUpperCase (f)), PChar (AnsiUpperCase ('k')))<>nil then
begin
        for b:=0 to (Length (g) -1) do
begin
            if (Ord (g[b])>47) and (Ord (g[b])<58) then
begin
                l:=1+g[b];
            end;
        end;

IRC_netChannelSettingsForm.IRC_netChannelSettings_Mode_l_SpinEdit.Value:=StrToIn
t (l);
        end
        else
        begin

IRC_netChannelSettingsForm.IRC_netChannelSettings_Mode_l_SpinEdit.Value:=StrToIn
t (Copy (g, 0, Length (g) -1));
        end;
    end;
    // +k
    if StrPos (PChar (AnsiUpperCase (f)), PChar (AnsiUpperCase ('k')))<>nil then
begin

```

```

IRC_netChannelSettingsForm.IRC_netChannelSettings_Mode_k_CheckBox.Checked:=true;
  if StrPos(PChar(AnsiUpperCase(f)),PChar(AnsiUpperCase('l')))<>nil then
    begin

IRC_netChannelSettingsForm.IRC_netChannelSettings_Mode_k_Edit.Text:=Copy(g,Length
h(1)+1,Length(g)-1);
    end
    else
    begin

IRC_netChannelSettingsForm.IRC_netChannelSettings_Mode_k_Edit.Text:=Copy(g,0,Len
gth(g)-1);
    end;
  end;
  // Показ режиму каналу в заголовку

IRC_netChannelSettingsForm.IRC_netChannelSettings_Mode_GroupBox.Caption:='MODE
'+d;
  end;

//if ACommand=IntToStr(329) then
TIRC_NETMemo (FindComponent(IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString(Copy(AContent,i+1,Length(AContent)-i));

// Немає топика на каналі
//if ACommand=IntToStr(331) then
TIRC_NETMemo (FindComponent(IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString(Copy(AContent,i+1,Length(AContent)-i));

// Ця команда вже оброблена в процедурі IdIRC_NETTopic
//if ACommand=IntToStr(332) then
TIRC_NETMemo (FindComponent(IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString(Copy(AContent,i+1,Length(AContent)-i));

if ACommand=IntToStr(341) then
TIRC_NETMemo (FindComponent(IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString(Copy(AContent,i+1,Length(AContent)-i));
if ACommand=IntToStr(342) then
TIRC_NETMemo (FindComponent(IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString(Copy(AContent,i+1,Length(AContent)-i));
if ACommand=IntToStr(351) then
TIRC_NETMemo (FindComponent(IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString(Copy(AContent,i+1,Length(AContent)-i));
if ACommand=IntToStr(352) then
TIRC_NETMemo (FindComponent(IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString(Copy(AContent,i+1,Length(AContent)-i));

// NAMES
if ACommand=IntToStr(353) then
  begin
    s:=Copy(AContent,i+3,Length(AContent)-i);
    n:=Pos(' ',s);
    if
AnsiUpperCase(IRC_net_PageControl.ActivePage.Caption)<>AnsiUpperCase(Copy(s,0,
n-1)) then

TIRC_NETMemo (FindComponent(IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString(Copy(AContent,i+3,Length(AContent)-i));
  end;
// End of /NAMES list.
if ACommand=IntToStr(366) then
  begin
    s:=Copy(AContent,i+1,Length(AContent)-i);
    n:=Pos(' ',s);
    if
AnsiUpperCase(IRC_net_PageControl.ActivePage.Caption)<>AnsiUpperCase(Copy(s,0,
n-1)) then

```

```

TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+' _IRC_NETMemo')).
AddFormattedString (Copy (AContent, i+1, Length (AContent) -i));
end;

if ACommand=IntToStr (364) then
TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+' _IRC_NETMemo')).
AddFormattedString (Copy (AContent, i+1, Length (AContent) -i));
if ACommand=IntToStr (365) then
TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+' _IRC_NETMemo')).
AddFormattedString (Copy (AContent, i+1, Length (AContent) -i));
// Список банів
if ACommand=IntToStr (367) then
begin
s:=Copy (AContent, i+1, Length (AContent) -i);
n:=Pos (' ', s);
d:=Copy (s, n+1, Length (s) -n);
m:=Pos (' ', d);
f:=Copy (d, 0, m-1);
if IRC_netChannelSettingsForm.Showing then
IRC_netChannelSettingsForm.IRC_netChannelSettings_Ban_ListBox.Items.Add (f)
else

TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+' _IRC_NETMemo')).
AddFormattedString (Copy (AContent, i+1, Length (AContent) -i));
end;
//if ACommand=IntToStr (368) then
TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+' _IRC_NETMemo')).
AddFormattedString (Copy (AContent, i+1, Length (AContent) -i));
if ACommand=IntToStr (369) then
TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+' _IRC_NETMemo')).
AddFormattedString (Copy (AContent, i+1, Length (AContent) -i));
if ACommand=IntToStr (371) then
TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+' _IRC_NETMemo')).
AddFormattedString (Copy (AContent, i+1, Length (AContent) -i));
if ACommand=IntToStr (372) then
TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+' _IRC_NETMemo')).
AddFormattedString (Copy (AContent, i+1, Length (AContent) -i));
if ACommand=IntToStr (374) then
TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+' _IRC_NETMemo')).
AddFormattedString (Copy (AContent, i+1, Length (AContent) -i));
if ACommand=IntToStr (375) then
TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+' _IRC_NETMemo')).
AddFormattedString (Copy (AContent, i+1, Length (AContent) -i));
if ACommand=IntToStr (376) then
TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+' _IRC_NETMemo')).
AddFormattedString (Copy (AContent, i+1, Length (AContent) -i));
if ACommand=IntToStr (381) then
TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+' _IRC_NETMemo')).
AddFormattedString (Copy (AContent, i+1, Length (AContent) -i));
if ACommand=IntToStr (382) then
TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+' _IRC_NETMemo')).
AddFormattedString (Copy (AContent, i+1, Length (AContent) -i));
if ACommand=IntToStr (391) then
TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+' _IRC_NETMemo')).
AddFormattedString (Copy (AContent, i+1, Length (AContent) -i));
if ACommand=IntToStr (392) then
TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+' _IRC_NETMemo')).
AddFormattedString (Copy (AContent, i+1, Length (AContent) -i));
if ACommand=IntToStr (393) then
TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+' _IRC_NETMemo')).
AddFormattedString (Copy (AContent, i+1, Length (AContent) -i));
if ACommand=IntToStr (394) then
TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+' _IRC_NETMemo')).
AddFormattedString (Copy (AContent, i+1, Length (AContent) -i));
if ACommand=IntToStr (395) then
TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+' _IRC_NETMemo')).
AddFormattedString (Copy (AContent, i+1, Length (AContent) -i));
end;

```

```

// Показувати отримані команди у вікні сервера
procedure TForm1.IdIRC_NETReceive(Sender: TObject; ACommand: String);
begin
  if IRC_netSettingsForm.IRC_netSettings_Show_Receive_Commands_CheckBox.Checked
  then
    IRC_net_Server_IRC_NETMemo.Lines.Add(ACommand);
end;

// Показувати відправлені команди у вікні сервера
procedure TForm1.IdIRC_NETSend(Sender: TObject; ACommand: String);
begin
  if IRC_netSettingsForm.IRC_netSettings_Show_Send_Commands_CheckBox.Checked
  then
    IRC_net_Server_IRC_NETMemo.Lines.Add('-> '+ACommand);
end;

// Прийом повідомлень у каналі й прийом особистих повідомлень
procedure TForm1.IdIRC_NETMessage(Sender: TObject; AUser: TIdIRC_NETUser;
  AChannel: TIdIRC_NETChannel; Content: String);
var
  i,n      : Integer;
begin
  // Дані про стан пінгу
  IRC_net_Ping_Timer.Enabled:=False;
  IRC_net_Ping_Timer.Enabled:=True;
  // Видалення рядків
  for i:=0 to IRC_net_PageControl.PageCount-1 do
    begin
      while
        TIRC_NETMemo (FindComponent (IRC_net_PageControl.Pages[i].Name+'_IRC_NETMemo')).Li
        nes.Count>IRC_netSettingsForm.IRC_netSettings_Servis_MaxLines_SpinEdit.Value do
          TIRC_NETMemo (FindComponent (IRC_net_PageControl.Pages[i].Name+'_IRC_NETMemo')).Li
          nes.Delete(0);
        end;
      // Прийом повідомлень у каналі
      try
        for i:=0 to IRC_net_PageControl.PageCount-1 do
          begin
            Application.ProcessMessages;
            if
              AnsiUpperCase (IRC_net_PageControl.Pages[i].Caption)=AnsiUpperCase (AChannel.Name)
            then
              begin
                TIRC_NETMemo (FindComponent (IRC_net_PageControl.Pages[i].Name+'_IRC_NETMemo')).Ad
                dFormattedString (TAG_COLOR+'1<'+AUser.Nick+'> '+Content);

                // Показати повідомлення
                if
                  IRC_netSettingsForm.IRC_netSettings_Show_Channels_Messages_CheckBox.Checked then
                    Begin
                      MSNPopUp.Title:='IRC_NET '+AChannel.Name+' '+AUser.Nick;
                      MSNPopUp.Text:=ParseColorString(Content);
                      MSNPopUp.GradientColor1:=$00FFCC99;
                      MSNPopUp.ShowPopUp;
                    end;

                // Робота з файлами лога
                IRC_NETLogs (AChannel.Name,AUser.Nick,Content);

                // Вивід повідомлення при згадуванні слів зі списку
                IRC_netFindText (AUser.Nick,Content,AChannel.Name);

                // Банити користувачів за використання слів зі списку
                IRC_netBotKick (AUser.Nick,Content,AChannel.Name);

                // Команди

```

```

IRC_netBotCommands (AUser.Nick,Content, AChannel.Name) ;

// Озвучити повідомлення в активному каналі
if
(IRC_netSettingsForm.IRC_netSettings_Sound_Speak_Channel_CheckBox.Checked) and
(AnsiUpperCase (IRC_net_PageControl.ActivePage.Caption)=AnsiUpperCase (AChannel.Name)) then
begin
    IRC_netSpeak (AUser.Nick,ParseColorString (Content) ,True) ;
end;

// Озвучити повідомлення у всіх каналах
if
IRC_netSettingsForm.IRC_netSettings_Sound_Speak_Channels_CheckBox.Checked then
begin
    IRC_netSpeak (AUser.Nick,ParseColorString (Content) ,True) ;
end;

// Звук
if
IRC_netSettingsForm.IRC_netSettings_Sound_Message_CheckBox.Checked then
begin
    Wave.WaveName:='message';
    Wave.Execute;
end;

// Іконка повідомлення
if IRC_net_PageControl.ActivePage.Caption<>AChannel.Name then
TTabSheet (FindComponent (IRC_net_PageControl.Pages [i].Name)).ImageIndex:=5;
end;
except
// Прийом особистих повідомлень
// Показати повідомлення
if IRC_netSettingsForm.IRC_netSettings_Show_Privat_Messages_CheckBox.Checked
then
begin
    MSNPopUp.Title:='IRC_NET '+AUser.Nick;
    MSNPopUp.Text:=ParseColorString (Content) ;
    MSNPopUp.GradientColor1:=$00D8BEDA;
    MSNPopUp.ShowPopUp;
end;

// Робота з файлами лога
IRC_NETLogs (AUser.Nick,AUser.Nick,Content) ;

// Озвучити повідомлення
if IRC_netSettingsForm.IRC_netSettings_Sound_Speak_Privat_CheckBox.Checked
then
begin
    IRC_netSpeak (AUser.Nick,ParseColorString (Content) ,False) ;
end;

for i:=0 to IRC_net_PageControl.PageCount-1 do
begin
    Application.ProcessMessages;
    if
AnsiUpperCase (IRC_net_PageControl.Pages [i].Caption)=AnsiUpperCase (AUser.Nick)
then
begin
    // Прийом особистих повідомлень до окремої вкладки
TIRC_NETMemo (FindComponent (IRC_net_PageControl.Pages [i].Name+'_IRC_NETMemo')).AddFormattedString (TAG_COLOR+'1<'+AUser.Nick+'> '+Content) ;

// Звук для особистого повідомлення

```

```

        if
IRC_netSettingsForm.IRC_netSettings_Sound_Privat_CheckBox.Checked then
        begin
            Wave.WaveName:='icq_o_o';
            Wave.Execute;
        end;

        // Іконка повідомлення
        if IRC_net_PageControl.ActivePage.Caption<>AUser.Nick then

TTabSheet (FindComponent (IRC_net_PageControl.Pages[i].Name)).ImageIndex:=6;
        end
        end;
        // Якщо вкладка привата з такою назвою вже існує, то зупиняємо процедуру
        for n:=0 to IRC_net_PageControl.PageCount-1 do
            begin
                if
AnsiUpperCase (IRC_net_PageControl.Pages[n].Caption)=AnsiUpperCase (AUser.Nick)
then
                begin
                    // Зупиняємо процедуру
                    Exit;
                end;
            end;
            // Створюємо нову вкладку й додаємо в неї повідомлення
            IRC_netNewPrivatPage (AUser.Nick,AUser.Nick,Content,6);

            // Звук для особистого повідомлення
            if IRC_netSettingsForm.IRC_netSettings_Sound_Privat_CheckBox.Checked then
                begin
                    Wave.WaveName:='icq_o_o';
                    Wave.Execute;
                end;

            // Прийом повідомлень у режимі AWAY
            if IdIRC_NET.Away then
                begin
                    for i:=0 to IRC_net_PageControl.PageCount-1 do
                        begin

IRC_net_Server_IRC_NETMemo.Lines.Add(IRC_net_PageControl.Pages[i].Caption+'=='+A
User.Nick);
                            if
AnsiUpperCase (IRC_net_PageControl.Pages[i].Caption)=AnsiUpperCase (AUser.Nick)
then
                                begin
                                    try
                                        IdIRC_NET.WriteLine ('PRIVMSG '+AUser.Nick+' :'+away_message);
                                    except
                                        on e:Exception do Reconnect (E.Message);
                                    end;

TIRC_NETMemo (FindComponent (IRC_net_PageControl.Pages[i].Name+'_IRC_NETMemo')).Ad
dFormattedString (TAG_COLOR+'1<'+IdIRC_NET.Nick+'> '+away_message);
                                    // Робота з файлами лога
                                    IRC_NETLogs (AUser.Nick,AUser.Nick,Content);
                                end;
                            end;
                        end;
                    end;
                end;
            end;

            // Прийом Action
            procedure TForm1.IdIRC_NETAction (Sender: TObject; AUser: TIdIRC_NETUser;
                AChannel: TIdIRC_NETChannel; Content: String);
            var
                i : Integer;
            begin
                for i:=0 to IRC_net_PageControl.PageCount-1 do

```

```

begin
  if
AnsiUpperCase(IRC_net_PageControl.Pages[i].Caption)=AnsiUpperCase(AChannel.Name)
then
  begin

TIRC_NETMemo(FindComponent(IRC_net_PageControl.Pages[i].Name+'_IRC_NETMemo')).Ad
dFormattedString(TAG_COLOR+'6* ' +AUser.Nick+' '+Content);

    // Показати повідомлення
    if
IRC_netSettingsForm.IRC_netSettings_Show_Channels_Messages_CheckBox.Checked then
      Begin
        MSNPopUp.Title:='IRC_NET '+AChannel.Name;
        MSNPopUp.Text:=ParseColorString(Content);
        MSNPopUp.GradientColor1:=$00FFCC99;
        MSNPopUp.ShowPopUp;
      end;

    // Вивід повідомлення при згадуванні слів зі списку
    IRC_netFindText(AUser.Nick,Content,AChannel.Name);

    // Банити користувачів за використання слів зі списку
    IRC_netBotKick(AUser.Nick,Content,AChannel.Name);

    // Озвучити повідомлення в активному каналі
    if
(IRC_netSettingsForm.IRC_netSettings_Sound_Speak_Channel_CheckBox.Checked) and
(AnsiUpperCase(IRC_net_PageControl.ActivePage.Caption)=AnsiUpperCase(AChannel.Na
me)) then
      begin
        IRC_netSpeak(AUser.Nick,ParseColorString(Content),True);
      end;

    // Озвучити повідомлення у всіх каналах
    if
IRC_netSettingsForm.IRC_netSettings_Sound_Speak_Channels_CheckBox.Checked then
      begin
        IRC_netSpeak(AUser.Nick,ParseColorString(Content),True);
      end;

    // Звук
    if IRC_netSettingsForm.IRC_netSettings_Sound_Message_CheckBox.Checked
then
      begin
        Wave.WaveName:='message';
        Wave.Execute;
      end;

    // Робота з файлами лога
    IRC_NETLogs(AChannel.Name,AUser.Nick,Content);

    // Іконка повідомлення
    if
AnsiUpperCase(IRC_net_PageControl.ActivePage.Caption)<>AnsiUpperCase(AChannel.Na
me) then

TTabSheet(FindComponent(IRC_net_PageControl.Pages[i].Name)).ImageIndex:=5;
      end;
    end;
  end;

procedure TForm1.IdIRC_NETKick(Sender: TObject; AUser, AVictim: TIdIRC_NETUser;
  AChannel: TIdIRC_NETChannel);
var
  i      : Integer;
begin
for i:=0 to IRC_net_PageControl.PageCount-1 do
  begin

```

```

    if
AnsiUpperCase(IRC_net_PageControl.Pages[i].Caption)=AnsiUpperCase(AChannel.Name)
then
    begin
        // Виводимо повідомлення

TIRC_NETMemo(FindComponent(IRC_net_PageControl.Pages[i].Name+'_IRC_NETMemo')).Ad
dFormattedString(TAG_COLOR+'3* '+AUser.Nick+' забанив користувача
'+AVictim.Nick+' ('+AVictim.Address+') - '+AUser.Reason);

        // Перевіряємо наявність ім'я користувача в списку
        if
TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Inde
xOf(AVictim.Nick)<>-1 then
            // Видаляємо ім'я користувача зі списку

TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Dele
te(TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.I
ndexOf(AVictim.Nick));

            // Якщо забанили оператора @
            // Перевіряємо наявність ім'я користувача в списку
            if
TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Inde
xOf('@'+AVictim.Nick)<>-1 then
                // Видаляємо ім'я користувача зі списку

TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Dele
te(TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.I
ndexOf('@'+AVictim.Nick));

                // Якщо забанили Voice +
                // Перевіряємо наявність ім'я користувача в списку
                if
TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Inde
xOf(''+AVictim.Nick)<>-1 then
                    // Видаляємо ім'я користувача зі списку

TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Dele
te(TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.I
ndexOf(''+AVictim.Nick));

                    // Якщо забанили Хопа %
                    // Перевіряємо наявність ім'я користувача в списку
                    if
TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Inde
xOf('%'+AVictim.Nick)<>-1 then
                        // Видаляємо ім'я користувача зі списку

TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Dele
te(TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.I
ndexOf('%'+AVictim.Nick));
                        end;
                    end;
                end;
            end;

// Вас забанили з каналу
procedure TForm1.IdIRC_NETKicked(Sender: TObject; AUser: TIdIRC_NETUser;
    AChannel: TIdIRC_NETChannel);
var
    i      : Integer;
begin
    for i:=0 to IRC_net_PageControl.PageCount-1 do
        begin
            if
AnsiUpperCase(IRC_net_PageControl.Pages[i].Caption)=AnsiUpperCase(AChannel.Name)
then
                begin

```



```

// Невідома команда
procedure TForm1.IdIRC_NETUnknownCommand(Sender: TObject; AUser: TIdIRC_NETUser;
  ACommand, AContent: String);
begin
  if AnsiUpperCase(ACommand)='ERROR' then
    begin
      Reconnect(AUser.Nick+' '+ACommand+' '+AContent);
    end
  else
    begin

TIRC_NETMemo(FindComponent(IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
Lines.Add(' = '+AUser.Nick+' '+ACommand+' '+AContent);
      end;
    end;

// Ваш статус
procedure TForm1.IdIRC_NETUserMode(Sender: TObject; AModes: String);
begin

TIRC_NETMemo(FindComponent(IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString(TAG_COLOR+'3* Ваш користувальницький статус змінений на
'+AModes);
    end;

// Новий користувач прийшов на канал
procedure TForm1.IdIRC_NETJoin(Sender: TObject; AUser: TIdIRC_NETUser;
  AChannel: TIdIRC_NETChannel);
var
  i      : Integer;
begin
  for i:=0 to IRC_net_PageControl.PageCount-1 do
    begin
      if
        AnsiUpperCase(IRC_net_PageControl.Pages[i].Caption)=AnsiUpperCase(AChannel.Name)
      then
        begin
          //Виводимо повідомлення
          if IRC_netSettingsForm.IRC_netSettings_Show_Join_CheckBox.Checked then
            begin

TIRC_NETMemo(FindComponent(IRC_net_PageControl.Pages[i].Name+'_IRC_NETMemo')).Ad
dFormattedString(TAG_COLOR+'3* '+AUser.Nick+' Заходить ('+AUser.Address+');
              end;

              // Перевіряємо наявність ім'я користувача в списку
              if
                TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Inde
xOf(AUser.Nick)=-1 then
                  // Додаємо користувача в список

TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Add(
AUser.Nick);

                  // Давати +o користувачам зі списку
                  IRC_netBotOp(AUser.Nick,AChannel.Name);

                  // Давати +v користувачам зі списку
                  IRC_netBotVoice(AUser.Nick,AChannel.Name);

                  // Звук
                  if IRC_netSettingsForm.IRC_netSettings_Sound_Join_CheckBox.Checked then
                    begin
                      Wave.WaveName:='join';
                      Wave.Execute;
                    end;
                end;
            end;
          end;
        end;
    end;
end;

```

```

// Перевірка наявності вікна привата з користувачем
for i:=0 to IRC_net_PageControl.PageCount-1 do
  begin
    if
AnsiUpperCase(IRC_net_PageControl.Pages[i].Caption)=AnsiUpperCase(AUser.Nick)
then
    begin
      // Міняємо іконку
      IRC_net_PageControl.Pages[i].ImageIndex:=7;
    end;
  end;
end;

// Користувач пішов з каналу
procedure TForm1.IdIRC_NETPart(Sender: TObject; AUser: TIdIRC_NETUser;
  AChannel: TIdIRC_NETChannel);
var
  i      : Integer;
begin
for i:=0 to IRC_net_PageControl.PageCount-1 do
  begin
    if
AnsiUpperCase(IRC_net_PageControl.Pages[i].Caption)=AnsiUpperCase(AChannel.Name)
then
    begin
      //Виводимо повідомлення
      if IRC_netSettingsForm.IRC_netSettings_Show_Part_CheckBox.Checked then
        begin
          TIRC_NETMemo(FindComponent(IRC_net_PageControl.Pages[i].Name+'_IRC_NETMemo')).AddFormattedString(TAG_COLOR+'3* '+AUser.Nick+' Вийшов ('+AUser.Address+')');
          end;

          // Перевіряємо наявність ім'я користувача в списку
          if
TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf(AUser.Nick)<>-1 then
            //Видаляємо користувача зі списку

TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Delete(TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf(AUser.Nick));

          // Якщо пішов оператор @
          // Перевіряємо наявність ім'я користувача в списку
          if
TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf('@'+AUser.Nick)<>-1 then
            //Видаляємо користувача зі списку

TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Delete(TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf('@'+AUser.Nick));

          // Якщо пішов Voice +
          // Перевіряємо наявність ім'я користувача в списку
          if
TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf(''+AUser.Nick)<>-1 then
            //Видаляємо користувача зі списку

TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Delete(TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf(''+AUser.Nick));

          // Якщо пішов Хоп %
          // Перевіряємо наявність ім'я користувача в списку

```

```

        if
TListBox (FindComponent (IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Inde
xOf('%'+AUser.Nick)<>-1 then
        //Видаляємо користувача зі списку

TListBox (FindComponent (IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Dele
te (TListBox (FindComponent (IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.I
ndexOf('%'+AUser.Nick));

        // Звук
        if IRC_netSettingsForm.IRC_netSettings_Sound_Part_CheckBox.Checked then
        begin
            Wave.WaveName:='part';
            Wave.Execute;
        end;
    end;
end;

// Приховання вкладки при виході з каналу
procedure TForm1.IdIRC_NETParted(Sender: TObject; AChannel: TIdIRC_NETChannel);
var
    i      : Integer;
begin
    for i:=0 to IRC_net_PageControl.PageCount-1 do
        begin
            if
AnsiUpperCase(IRC_net_PageControl.Pages[i].Caption)=AnsiUpperCase(AChannel.Name)
then
                begin
                    IRC_net_PageControl.Pages[i].TabVisible:=false;
                end;
        end;
    end;

// Користувач відключився (QUIT)
procedure TForm1.IdIRC_NETQuit(Sender: TObject; AUser: TIdIRC_NETUser);
var
    i : Integer;
begin
    // Повідомлення в канали де був користувач
    for i:=0 to IRC_net_PageControl.PageCount-1 do
        begin
            if IRC_net_PageControl.Pages[i].Caption[1]='#' then
                begin
                    // Перевіряємо наявність ім'я користувача в списку
                    if
TListBox (FindComponent (IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Inde
xOf(AUser.Nick)<>-1 then
                        begin
                            //Видаляємо користувача зі списку

TListBox (FindComponent (IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Dele
te (TListBox (FindComponent (IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.I
ndexOf(AUser.Nick));
                            // Виводимо повідомлення
                            if IRC_netSettingsForm.IRC_netSettings_Show_Quit_CheckBox.Checked
then
                                begin

TIRC_NETMemo (FindComponent (IRC_net_PageControl.Pages[i].Name+'_IRC_NETMemo')).Ad
dFormattedString(TAG_COLOR+'2* '+AUser.Nick+' Відключився ('+AUser.Address+') -
'+AUser.Reason);
                                    end;
                                end;

                            // Якщо пішов оператор @
                            // Перевіряємо наявність ім'я користувача в списку

```

```

        if
TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Inde
xOf('@'+AUser.Nick)<>-1 then
        begin
            //Видаляємо користувача зі списку

TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Dele
te(TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.I
ndexOf('@'+AUser.Nick));
            // Виводимо повідомлення
            if IRC_netSettingsForm.IRC_netSettings_Show_Quit_CheckBox.Checked
then
                begin

TIRC_NETMemo(FindComponent(IRC_net_PageControl.Pages[i].Name+'_IRC_NETMemo')).Ad
dFormattedString(TAG_COLOR+'2* '+AUser.Nick+' Відключився ('+AUser.Address+') -
'+AUser.Reason);
                    end;
                end;

                // Якщо пішов Voice +
                // Перевіряємо наявність ім'я користувача в списку
                if
TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Inde
xOf('+'+AUser.Nick)<>-1 then
                    begin
                        //Видаляємо користувача зі списку

TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Dele
te(TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.I
ndexOf('+'+AUser.Nick));
                            // Виводимо повідомлення
                            if IRC_netSettingsForm.IRC_netSettings_Show_Quit_CheckBox.Checked
then
                                begin

TIRC_NETMemo(FindComponent(IRC_net_PageControl.Pages[i].Name+'_IRC_NETMemo')).Ad
dFormattedString(TAG_COLOR+'2* '+AUser.Nick+' Відключився ('+AUser.Address+') -
'+AUser.Reason);
                                    end;
                                end;

                                // Якщо пішов Хоп %
                                // Перевіряємо наявність ім'я користувача в списку
                                if
TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Inde
xOf('%'+AUser.Nick)<>-1 then
                                    begin
                                        //Видаляємо користувача зі списку

TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Dele
te(TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.I
ndexOf('%'+AUser.Nick));
                                            // Виводимо повідомлення
                                            if IRC_netSettingsForm.IRC_netSettings_Show_Quit_CheckBox.Checked
then
                                                begin

TIRC_NETMemo(FindComponent(IRC_net_PageControl.Pages[i].Name+'_IRC_NETMemo')).Ad
dFormattedString(TAG_COLOR+'2* '+AUser.Nick+' Відключився ('+AUser.Address+') -
'+AUser.Reason);
                                                    end;
                                                end;
                                                end;
                                                end;

// Повідомлення у вікно привата з користувачем
for i:=0 to IRC_net_PageControl.PageCount-1 do
        begin

```

```

    if
AnsiUpperCase(IRC_net_PageControl.Pages[i].Caption)=AnsiUpperCase(AUser.Nick)
then
    begin
        if IRC_netSettingsForm.IRC_netSettings_Show_Quit_CheckBox.Checked then
            begin

TIRC_NETMemo(FindComponent(IRC_net_PageControl.Pages[i].Name+'_IRC_NETMemo')).Ad
dFormattedString(TAG_COLOR+'2* '+AUser.Nick+' Відключився ('+AUser.Address+') -
'+AUser.Reason);
                end;
                // Міняємо іконку
                IRC_net_PageControl.Pages[i].ImageIndex:=16;
            end;
        end;
end;

// Користувач міняє нік
procedure TForm1.IdIRC_NETNickChange(Sender: TObject; AUser: TIdIRC_NETUser;
    ANewNick: String);
var
    i : integer;
begin
for i:=0 to IRC_net_PageControl.PageCount-1 do
    begin
        // Зміна ім'я вкладки для привата
        if
AnsiUpperCase(IRC_net_PageControl.Pages[i].Caption)=AnsiUpperCase(AUser.Nick)
then IRC_net_PageControl.Pages[i].Caption:=ANewNick;

            if IRC_net_PageControl.Pages[i].Caption[1]='#' then
                begin
                    // Якщо перемінив нік звичайний користувач
                    // Перевіряємо наявність ім'я користувача в списку
                    if
TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Inde
xOf(AUser.Nick)<>-1 then
                        begin
                            //Видаляємо користувача зі списку

TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Dele
te(TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.I
ndexOf(AUser.Nick));
                            // Додаємо користувача в список з новим ніком

TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Add(
ANewNick);
                            // Виводимо повідомлення
                            if IRC_netSettingsForm.IRC_netSettings_Show_Nick_CheckBox.Checked
then
                                begin

TIRC_NETMemo(FindComponent(IRC_net_PageControl.Pages[i].Name+'_IRC_NETMemo')).Ad
dFormattedString(TAG_COLOR+'3* '+AUser.Nick+' міняє нік на '+ANewNick);
                                    end;
                                    end;

                                // Якщо перемінив нік оператор @
                                // Перевіряємо наявність ім'я користувача в списку
                                if
TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Inde
xOf('@'+AUser.Nick)<>-1 then
                                    begin
                                        //Видаляємо користувача зі списку

TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Dele
te(TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.I
ndexOf('@'+AUser.Nick));
                                        // Додаємо користувача в список з новим ніком

```

```

TListBox (FindComponent (IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Add(
'@'+ANewNick);
    // Виводимо повідомлення
    if IRC_netSettingsForm.IRC_netSettings_Show_Nick_CheckBox.Checked
then
    begin

TIRC_NETMemo (FindComponent (IRC_net_PageControl.Pages[i].Name+'_IRC_NETMemo')).Ad
dFormattedString (TAG_COLOR+'3* '+AUser.Nick+' міняє нік на '+ANewNick);
    end;
    end;

    // Якщо перемінив нік Voice +
    // Перевіряємо наявність ім'я користувача в списку
    if
TListBox (FindComponent (IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Inde
xOf (''+AUser.Nick)<>-1 then
    begin
        //Видаляємо користувача зі списку

TListBox (FindComponent (IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Dele
te (TListBox (FindComponent (IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.I
ndexOf (''+AUser.Nick));
        // Додаємо користувача в список з новим ніком

TListBox (FindComponent (IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Add(
''+ANewNick);
        // Виводимо повідомлення
        if IRC_netSettingsForm.IRC_netSettings_Show_Nick_CheckBox.Checked
then
            begin

TIRC_NETMemo (FindComponent (IRC_net_PageControl.Pages[i].Name+'_IRC_NETMemo')).Ad
dFormattedString (TAG_COLOR+'3* '+AUser.Nick+' міняє нік на '+ANewNick);
            end;
            end;

            // Якщо перемінив нік Хоп %
            // Перевіряємо наявність ім'я користувача в списку
            if
TListBox (FindComponent (IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Inde
xOf ('%'+AUser.Nick)<>-1 then
                begin
                    //Видаляємо користувача зі списку

TListBox (FindComponent (IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Dele
te (TListBox (FindComponent (IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.I
ndexOf ('%'+AUser.Nick));
                    // Додаємо користувача в список з новим ніком

TListBox (FindComponent (IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Add(
'%'+ANewNick);
                    // Виводимо повідомлення
                    if IRC_netSettingsForm.IRC_netSettings_Show_Nick_CheckBox.Checked
then
                        begin

TIRC_NETMemo (FindComponent (IRC_net_PageControl.Pages[i].Name+'_IRC_NETMemo')).Ad
dFormattedString (TAG_COLOR+'3* '+AUser.Nick+' міняє нік на '+ANewNick);
                        end;
                        end;

                        end;
                        // Перевірка наявності вікна привата з користувачем
                        for i:=0 to IRC_net_PageControl.PageCount-1 do
                            begin
                                if
AnsiUpperCase (IRC_net_PageControl.Pages[i].Caption)=AnsiUpperCase (ANewNick) then

```

```

begin
    // Міняємо іконку
    IRC_net_PageControl.Pages[i].ImageIndex:=7;
end;
end;

// Ви міняєте нік
procedure TForm1.IdIRC_NETNickChanged(Sender: TObject; AOldNick: String);
var
    i : integer;
begin
    IRC_net_Nik_Edit.Text:=IdIRC_NET.Nick;
    for i:=0 to IRC_net_PageControl.PageCount-1 do
        begin
            if IRC_net_PageControl.Pages[i].Caption[1]='#' then
                begin
                    // Якщо ви звичайний користувач
                    // Перевіряємо наявність ім'я користувача в списку
                    if
                        TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf(AOldNick)<>-1 then
                            begin
                                //Видаляємо користувача зі списку

                                TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Delete(TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf(AOldNick));
                                // Додаємо користувача в список з новим ніком

                                TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Add(IdIRC_NET.Nick);
                                // Виводимо повідомлення

                                TIRC_NETMemo(FindComponent(IRC_net_PageControl.Pages[i].Name+'_IRC_NETMemo')).AddFormattedString(TAG_COLOR+'3* Ваш новий нік '+IdIRC_NET.Nick);
                                end;

                                // Якщо ви оператор @
                                // Перевіряємо наявність ім'я користувача в списку
                                if
                                    TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf('@'+AOldNick)<>-1 then
                                        begin
                                            //Видаляємо користувача зі списку

                                            TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Delete(TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf('@'+AOldNick));
                                            // Додаємо користувача в список з новим ніком

                                            TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Add('@'+IdIRC_NET.Nick);
                                            // Виводимо повідомлення

                                            TIRC_NETMemo(FindComponent(IRC_net_PageControl.Pages[i].Name+'_IRC_NETMemo')).AddFormattedString(TAG_COLOR+'3* Ваш новий нік '+IdIRC_NET.Nick);
                                            end;

                                            // Якщо ви Voice +
                                            // Перевіряємо наявність ім'я користувача в списку
                                            if
                                                TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf(''+AOldNick)<>-1 then
                                                    begin
                                                        //Видаляємо користувача зі списку

                                                        TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Delete

```

```

te(TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Ind
indexOf(''+AOldNick));
    // Додаємо користувача в список з новим ніком

TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Add(
'+'+IdIRC_NET.Nick);
    // Виводимо повідомлення

TIRC_NETMemo(FindComponent(IRC_net_PageControl.Pages[i].Name+'_IRC_NETMemo')).Ad
dFormattedString(TAG_COLOR+'3* Ваш новий нік '+IdIRC_NET.Nick);
    end;

    // Якщо ви Хоп %
    // Перевіряємо наявність ім'я користувача в списку
    if
TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Inde
xOf('%'+AOldNick)<>-1 then
    begin
        //Видаляємо користувача зі списку

TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Dele
te(TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.I
ndexOf('%'+AOldNick));
        // Додаємо користувача в список з новим ніком

TListBox(FindComponent(IRC_net_PageControl.Pages[i].Name+'_ListBox')).Items.Add(
'+'+IdIRC_NET.Nick);
        // Виводимо повідомлення

TIRC_NETMemo(FindComponent(IRC_net_PageControl.Pages[i].Name+'_IRC_NETMemo')).Ad
dFormattedString(TAG_COLOR+'3* Ваш новий нік '+IdIRC_NET.Nick);
        end;
    end;
end;

// Нік і AltNick перебувають у використанні.
procedure TForm1.IdIRC_NETNicksInUse(Sender: TObject; var ANick: String);
begin
    IdIRC_NET.Disconnect;

TIRC_NETMemo(FindComponent(IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString(TAG_COLOR+'4Ваші Ніки '+IdIRC_NET.Nick+' і
'+IdIRC_NET.AltNick+' уже використовуються іншими
користувачами.'+#10#13+'Перемініте нік і підключитесь заново.');
```

```

        on e:Exception do Reconnect(E.Message);
    end;
end;

// Перезахід на відкриті канали
if (ACmdCode=255) and
(IRC_netSettingsForm.IRC_netSettings_Servis_Quit_CheckBox.Checked) then
begin
    for i:=0 to IRC_net_PageControl.PageCount-1 do
        begin
            if IRC_net_PageControl.Pages[i].Caption[1]='#' then
                begin
                    try
                        IdIRC_NET.Join(IRC_net_PageControl.Pages[i].Caption);

TIRC_NETMemo (FindComponent(IRC_net_PageControl.Pages[i].Name+'_IRC_NETMemo')).AddFormattedString(TAG_COLOR+'4* Перезахід на канал '+IRC_net_PageControl.Pages[i].Caption);
                    except
                        // Помилка
                        on e:Exception do
TIRC_NETMemo (FindComponent(IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).AddFormattedString(TAG_COLOR+'4'+E.Message);
                    end;
                end;
            end;
        end;
    end;

// Автозахід при з'єднанні
if (ACmdCode=255) and (IRC_netFavoriteForm.IRC_netFavorite_CheckBox.Checked) and
(IRC_net_PageControl.PageCount=2) then
begin
    FavoriteList:=TStringList.Create;
    try

FavoriteList.LoadFromFile(ExtractFileDir(ParamStr(0))+'\IRC_NET\Favorite\'+'+IRC_net_Server_Address_Combobox.Text+'.fav');
        for i:=0 to FavoriteList.Count-1 do
            begin
                IdIRC_NET.Join(FavoriteList.Strings[i]);
            end;
        except
            // Помилка
            on e:Exception do
TIRC_NETMemo (FindComponent(IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).AddFormattedString(TAG_COLOR+'4'+E.Message);
            end;
            FavoriteList.Free;
        end;
    end;

// Користувач надіслав Вам запит STCP
procedure TForm1.IdIRC_NETSTCPQuery(Sender: TObject; User: TIdIRC_NETUser;
    AChannel: TIdIRC_NETChannel; Command, Args: String; var ASuppress: Boolean);
begin

TIRC_NETMemo (FindComponent(IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).AddFormattedString(TAG_COLOR+'12* ['+User.Nick+'] відправив Вам STCP запит : '+Command+' '+Args);
end;

// Отримана відповідь на запит STCP
procedure TForm1.IdIRC_NETSTCPReply(Sender: TObject; AUser: TIdIRC_NETUser;
    AChannel: TIdIRC_NETChannel; Command, Args: String);
begin

TIRC_NETMemo (FindComponent(IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).AddFormattedString(TAG_COLOR+'12* ['+AUser.Nick+'] відповідь на STCP запит : '+Command+' '+Args);
end;

```

```

end;

// Прийом повідомлень (Notice)
procedure TForm1.IdIRC_NETNotice(Sender: TObject; AUser: TIdIRC_NETUser;
  AChannel: TIdIRC_NETChannel; Content: String);
var
  i      : Integer;
begin
  try
    for i:=0 to IRC_net_PageControl.PageCount-1 do
      begin
        if
          AnsiUpperCase(IRC_net_PageControl.Pages[i].Caption)=AnsiUpperCase(AChannel.Name)
        then
          begin
            TIRC_NETMemo(FindComponent(IRC_net_PageControl.Pages[i].Name+'_IRC_NETMemo')).AddFormattedString(TAG_COLOR+'7* - '+AUser.Nick+' - '+Content);
            end;
          end;
        except
          TIRC_NETMemo(FindComponent(IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).AddFormattedString(TAG_COLOR+'7* - '+AUser.Nick+' - '+Content);
          end;
        end;

// Режим AWAY включений
procedure TForm1.IdIRC_NETNowAway(Sender: TObject; AContent: String);
begin
  IRC_net_Away_Status_Label.Visible:=true;
end;

// Назва топика на каналі
procedure TForm1.IdIRC_NETTopic(Sender: TObject; AUser: TIdIRC_NETUser;
  AChannel: TIdIRC_NETChannel; const AChanName, ATopic: String);
var
  i      : Integer;
  LogFile : TextFile;
  TopicList : TStringList;
begin
  for i:=0 to IRC_net_PageControl.PageCount-1 do
    begin
      if
        AnsiUpperCase(IRC_net_PageControl.Pages[i].Caption)=AnsiUpperCase(AChannel.Name)
      then
        begin
          TIRC_NETMemo(FindComponent(IRC_net_PageControl.Pages[i].Name+'_IRC_NETMemo')).AddFormattedString(TAG_COLOR+'7* Топік '+AUser.Nick+' '+ATopic);

          // Збереження топика у файл

          AssignFile(LogFile, ExtractFileDir(ParamStr(0))+'\IRC_NET\Channel_Settings\Topic\'+IdIRC_NET.Host+'\'+AChannel.Name+'.txt');
          if
            FileExists(ExtractFilePath(paramstr(0))+'\IRC_NET\Channel_Settings\Topic\'+IdIRC_NET.Host+'\'+AChannel.Name+'.txt')=false then
              begin
                CreateDirectory(PChar(ExtractFileDir(ParamStr(0))+'\IRC_NET'), nil);

                CreateDirectory(PChar(ExtractFileDir(ParamStr(0))+'\IRC_NET\Channel_Settings'), nil);

                CreateDirectory(PChar(ExtractFileDir(ParamStr(0))+'\IRC_NET\Channel_Settings\Topic'), nil);
              end;
        end;
    end;
  end;

```

```

CreateDirectory(PChar(ExtractFileDir(ParamStr(0))+'\IRC_NET\Channel_Settings\Topic\'+IdIRC_NET.Host),nil);
    Rewrite(LogFile);
    CloseFile(LogFile);
end;

    TopicList:=TStringList.Create;

TopicList.LoadFromFile(ExtractFilePath(paramstr(0))+'\IRC_NET\Channel_Settings\Topic\'+IdIRC_NET.Host+'\'+AChannel.Name+'.txt');
    if (TopicList.IndexOf(ATopic)=-1) and (ATopic<>') then
    begin
        Append(LogFile);
        Writeln(LogFile,ATopic);
        TopicList.Free;
        CloseFile(LogFile);
    end;
end;
end;

// Режим AWAY виключений
procedure TForm1.IdIRC_NETUnAway(Sender: TObject; AContent: String);
begin
    IRC_net_Away_Status_Label.Visible:=false;
end;

procedure TForm1.IdIRC_NETPingPong(Sender: TObject);
begin
    // Дані про стан пінга
    IRC_net_Ping_Timer.Enabled:=False;
    IRC_net_Ping_Timer.Enabled:=True;

    if IRC_netSettingsForm.IRC_netSettings_Show_PingPong_CheckBox.Checked then
        IRC_net_Server_IRC_NETMemo.AddFormattedString(TAG_COLOR+'3* Ping! Pong!'
'+TimeToStr(Time));
end;

// помилка сокету
procedure TForm1.IdIRC_NETSocketError(Sender: TObject; ASocket, AMsg: String);
begin

TIRC_NETMemo(FindComponent(IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString(TAG_COLOR+'4=SocketError='+ASocket+' '+AMsg);
end;

// помилка
procedure TForm1.IdIRC_NETError(Sender: TObject; AUser: TIdIRC_NETUser;
ANumeric,
    AError: String);
begin

TIRC_NETMemo(FindComponent(IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString(TAG_COLOR+'4=Error='+AUser.Nick+' '+ANumeric+' '+AError);
end;

// Користувач запросив вас на канал
procedure TForm1.IdIRC_NETInvite(Sender: TObject; AUser: TIdIRC_NETUser;
    AChannel: String);
begin
TIRC_NETMemo(FindComponent(IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString(TAG_COLOR+'6* '+AUser.Nick+' запрошує вас на канал
'+AChannel);

// Заходити в канал при запрошенні
if IRC_netSettingsForm.IRC_netSettings_Servis_Invite_CheckBox.Checked then
begin
    try

```

```

        IdIRC_NET.Join(AChannel);
    except
        // Помилка
        on e:Exception do
TIRC_NETMemo (FindComponent(IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString(TAG_COLOR+'4'+E.Message);
        end;
    end;
end;

// Дії при показі вкладок
// Процедура показу вкладки вікна сервера
procedure TForm1.IRC_net_ServerShow(Sender: TObject);
begin
    IRC_net_Close_SpeedButton.Enabled:=false;
    IRC_net_Close_SpeedButton.ShowHint:=false;
    try
        // Міняємо іконку для вікна каналу
        if IRC_net_PageControl.ActivePage.ImageIndex=5 then
IRC_net_PageControl.ActivePage.ImageIndex:=4;
        // Міняємо іконку для вікна привата
        if IRC_net_PageControl.ActivePage.ImageIndex=6 then
IRC_net_PageControl.ActivePage.ImageIndex:=7;
        except
            end;
    end;

// Процедура показу вкладки списку каналів
procedure TForm1.ChannelsShow(Sender: TObject);
begin
    IRC_net_Close_SpeedButton.Enabled:=true;
    IRC_net_Close_SpeedButton.ShowHint:=true;
    try
        // Міняємо іконку для вікна каналу
        if IRC_net_PageControl.ActivePage.ImageIndex=5 then
IRC_net_PageControl.ActivePage.ImageIndex:=4;
        // Міняємо іконку для вікна привата
        if IRC_net_PageControl.ActivePage.ImageIndex=6 then
IRC_net_PageControl.ActivePage.ImageIndex:=7;
        except
            end;
    end;

// Процедура показу вкладки каналу
procedure TForm1.IRC_net_Channel_Show(Sender: TObject);
begin
    IRC_net_Close_SpeedButton.Enabled:=true;
    IRC_net_Close_SpeedButton.ShowHint:=true;
    try
        // Міняємо іконку для вікна каналу
        if IRC_net_PageControl.ActivePage.ImageIndex=5 then
IRC_net_PageControl.ActivePage.ImageIndex:=4;
        // Міняємо іконку для вікна привата
        if IRC_net_PageControl.ActivePage.ImageIndex=6 then
IRC_net_PageControl.ActivePage.ImageIndex:=7;
        except
            end;
    end;

// Процедура показу вкладки привата
procedure TForm1.IRC_net_Privat_Show(Sender: TObject);
begin
    IRC_net_Close_SpeedButton.Enabled:=true;
    IRC_net_Close_SpeedButton.ShowHint:=true;
    try
        // Міняємо іконку для вікна каналу
        if IRC_net_PageControl.ActivePage.ImageIndex=5 then
IRC_net_PageControl.ActivePage.ImageIndex:=4;
        // Міняємо іконку для вікна привата

```

```

        if IRC_net_PageControl.ActivePage.ImageIndex=6 then
IRC_net_PageControl.ActivePage.ImageIndex:=7;
        except
        end;
end;

// Кнопки
// Очистити вікно показу повідомлень
procedure TForm1.IRC_net_Clear_SpeedButtonClick(Sender: TObject);
begin

TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
Clear;
end;

// Показати балка каналу
procedure TForm1.IRC_net_Log_SpeedButtonClick(Sender: TObject);
begin
    ShellExecute
(0, 'open', Pchar (ExtractFileDir (ParamStr (0)) + '\IRC_NET\Logs\' + IdIRC_NET.Host + '\'+
IRC_net_PageControl.ActivePage.Caption + '.txt'), '', '', sw_show);
end;

// Вибране
procedure TForm1.IRC_net_Favorite_SpeedButtonClick(Sender: TObject);
var
    item          : TMenuItem;
    i              : integer;
    FavoriteList  : TStringList;
begin
for i:=0 to IRC_net_Favorite_PopupMenu.Items.Count-1 do
begin
    if IRC_net_Favorite_PopupMenu.Items.Count-1>2 then
IRC_net_Favorite_PopupMenu.Items[IRC_net_Favorite_PopupMenu.Items.Count-1].Free;
end;
if
FileExists (ExtractFileDir (ParamStr (0)) + '\IRC_NET\Favorite\' + IRC_net_Server_Addre
ss_Combobox.Text + '.fav')=true then
begin
    FavoriteList:=TStringList.Create;

FavoriteList.LoadFromFile (ExtractFileDir (ParamStr (0)) + '\IRC_NET\Favorite\' + IRC_n
et_Server_Address_Combobox.Text + '.fav');
    FavoriteList.Sorted:=true;
    for i:=0 to FavoriteList.Count-1 do
begin
        item:=TMenuItem.Create (IRC_net_Favorite_PopupMenu);
        item.Caption:=FavoriteList.Strings[i];
        item.OnClick:=ItemClick;
        IRC_net_Favorite_PopupMenu.Items.Add (item);
end;
        FavoriteList.Free;
end;
IRC_net_Favorite_PopupMenu.Popup (Mouse.CursorPos.X, Mouse.CursorPos.Y);
end;

// Клік по пункті ім'я каналу меню вибраного й захід на канал
procedure TForm1.ItemClick (sender: TObject);
begin
if IdIRC_NET.Connected then
begin
    try
        with sender as TMenuItem do
begin
            IdIRC_NET.Join (Caption);
end;
        except
            // Помилка

```

```

        on e:Exception do
TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString (TAG_COLOR+'4'+E.Message);
        end;
        end;
end;

// Додати у вибране
procedure TForm1.IRC_net_Favorite_PopupMenu_AddClick(Sender: TObject);
var
    i      : Integer;
    // Робота з файлами Вибраного
    LogFile : TextFile;
begin
for i:=0 to IRC_net_Favorite_PopupMenu.Items.Count-1 do
    begin
        if
AnsiUpperCase (IRC_net_PageControl.ActivePage.Caption)=AnsiUpperCase (IRC_net_Favo
rite_PopupMenu.Items[i].Caption) then exit;
        end;
if IRC_net_PageControl.ActivePage.Caption[1]='#' then
    begin
        // Робота з файлами Вибраного

AssignFile (LogFile, ExtractFileDir (ParamStr (0))+'\IRC_NET\Favorite\'+IdIRC_NET.Ho
st+'.fav');
        if
FileExists (ExtractFileDir (ParamStr (0))+'\IRC_NET\Favorite\'+IdIRC_NET.Host+'.fav
')=false then
            begin
                CreateDirectory (PChar (ExtractFileDir (ParamStr (0))+'\IRC_NET'), nil);

CreateDirectory (PChar (ExtractFileDir (ParamStr (0))+'\IRC_NET\Favorite'), nil);
                Rewrite (LogFile);
                CloseFile (LogFile);
            end;
            Append (LogFile);
            Writeln (LogFile, IRC_net_PageControl.ActivePage.Caption);
            CloseFile (LogFile);
        end;
    end;

// Упорядкувати вибране
procedure TForm1.IRC_net_Favorite_PopupMenu_DelClick(Sender: TObject);
begin
    IRC_netFavoriteForm.ShowModal;
end;

// Команди
procedure TForm1.IRC_net_Commands_SpeedButtonClick(Sender: TObject);
begin
if IdIRC_NET.Connected then
    begin
        IRC_net_Commands_PopupMenu.Popup (Mouse.CursorPos.X, Mouse.CursorPos.Y);
    end
    else
        begin
TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString (TAG_COLOR+'4Ви не підключені до IRC_NET сервера');
        end;
    end;

// Вікно властивостей каналу
procedure TForm1.IRC_net_ChannelSettings_SpeedButtonClick(Sender: TObject);
begin
    if not IdIRC_NET.Connected then
        begin

```

```

TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString (TAG_COLOR+'4Ви не підключені до IRC_NET сервера');
    exit;
end;
if IRC_net_PageControl.ActivePage.Caption[1]='#' then
begin
    IRC_netChannelSettingsForm.Caption:='Властивості каналу
'+IRC_net_PageControl.ActivePage.Caption;
    IRC_netChannelSettingsForm.ShowModal;
end
else
begin
TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString (TAG_COLOR+'4Перемкнетесь на вкладку каналу');
end;
end;

// Меню списку каналів
procedure TForm1.IRC_net_Channels_SpeedButtonClick(Sender: TObject);
begin
if IdIRC_NET.Connected then
begin
    IRC_net_List_PopupMenu.Popup (Mouse.CursorPos.X,Mouse.CursorPos.Y);
end
else
begin
TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString (TAG_COLOR+'4Ви не підключені до IRC_NET сервера');
end;
end;

// Більше 100 користувачів на каналах
procedure TForm1.IRC_net_List_100_PopupMenuClick(Sender: TObject);
begin
if IdIRC_NET.Connected then
begin
    IRC_net_PageControl.ActivePageIndex:=1;
    try
        // Відправляємо повідомлення серверу
        IdIRC_NET.WriteLine ('LIST >100');
    except
        on e:Exception do Reconnect (E.Message);
    end;
end
else
begin
TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString (TAG_COLOR+'4Ви не підключені до IRC_NET сервера');
end;
end;

// Більше 50 користувачів на каналах
procedure TForm1.IRC_net_List_50_PopupMenuClick(Sender: TObject);
begin
if IdIRC_NET.Connected then
begin
    IRC_net_PageControl.ActivePageIndex:=1;
    try
        // Відправляємо повідомлення серверу
        IdIRC_NET.WriteLine ('LIST >50');
    except
        on e:Exception do Reconnect (E.Message);
    end;
end
else

```

```

begin

TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString (TAG_COLOR+'4Ви не підключені до IRC_NET сервера');
end;
end;

// Всі канали
procedure TForm1.IRC_net_List_All_PopupMenuClick(Sender: TObject);
begin
if IdIRC_NET.Connected then
begin
IRC_net_PageControl.ActivePageIndex:=1;
try
// Відправляємо повідомлення серверу
IdIRC_NET.WriteLine('LIST');
except
on e:Exception do Reconnect(E.Message);
end;
end
else
begin

TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString (TAG_COLOR+'4Ви не підключені до IRC_NET сервера');
end;
end;

// Показ вікна налаштувань IRC_NET бота
procedure TForm1.IRC_net_Bot_SpeedButtonClick(Sender: TObject);
begin
IRC_netBotForm.ShowModal;
end;

// Показ вікна налаштувань IRC NET
procedure TForm1.IRC_net_Settings_SpeedButtonClick(Sender: TObject);
begin
IRC_netSettingsForm.ShowModal;
end;

// Показ Загального FAQ по IRC NET
procedure TForm1.IRC_net_Faq_SpeedButtonClick(Sender: TObject);
var
Res : TResourceStream;
begin
Res:= TResourceStream.Create(HInstance, 'IRC_NETHELP', 'HTML');
Res.SaveToFile (ExtractFileDir (ParamStr(0)) + '\ ' + 'IRC_net_faq.html');
Res.Free;
ShellExecute (Application.handle, 'open', pchar (ExtractFileDir (ParamStr(0)) +
'\ ' + 'IRC_net_faq.html'), nil, nil, SW_SHOW); //відкрили в браузері за
замовчуванням
TIdAntiFreezeBase.Sleep(10000);
// Видаляємо файл IRC_net_faq.html
DeleteFile (ExtractFileDir (ParamStr(0)) + '\ ' + 'IRC_net_faq.html');
end;

// Закриття вкладки каналу й вихід з каналу, приховання вкладки списку каналів,
закриття вікна привата
procedure TForm1.IRC_net_Close_SpeedButtonClick(Sender: TObject);
begin
// Приховуємо вкладку списку каналів
if IRC_net_PageControl.ActivePage.Name='Channels' then
begin
Channels.TabVisible:=false;
end
else
begin
if IRC_net_PageControl.ActivePage.Caption[1]='#' then
begin

```

```

        if IdIRC_NET.Connected then
            begin
                try
                    // Відправляємо повідомлення серверу
                    IdIRC_NET.WriteLine('PART
'+IRC_net_PageControl.ActivePage.Caption);
                except
                    on e:Exception do Reconnect(E.Message);
                end;
            end;
            // Закриваємо вкладку каналу
            IRC_net_PageControl.ActivePage.Free;
        end
    else
        begin
            // Закриваємо вкладку привата
            IRC_net_PageControl.ActivePage.Free;
        end;
    end;
end;

// Відправлення повідомлень у канал і команд серверу
procedure TForm1.IRC_net_Channels_MessageText_EditKeyPress(Sender: TObject;
    var Key: Char);
var
    i,c      : Integer;
    s        : String;
    // Створення вкладки для привата - команда QUERY
    ip,np    : Integer;
begin
    i:=Pos('
',TEdit(FindComponent(IRC_net_PageControl.ActivePage.Name+'_Edit')).Text);
    s:=Copy(TEdit(FindComponent(IRC_net_PageControl.ActivePage.Name+'_Edit')).Text,i+1,Length(TEdit(FindComponent(IRC_net_PageControl.ActivePage.Name+'_Edit')).Text)-i);
    c:=Pos(' ',s);
    if (key = #13) and (idIRC_NET.Connected) then
        begin
            if
                TEdit(FindComponent(IRC_net_PageControl.ActivePage.Name+'_Edit')).Text<>' ' then
                    begin
                        try
                            if
                                TEdit(FindComponent(IRC_net_PageControl.ActivePage.Name+'_Edit')).Text[1]='/'
                            then
                                begin
                                    // me (Action)
                                    if
                                        (AnsiUpperCase(Copy(TEdit(FindComponent(IRC_net_PageControl.ActivePage.Name+'_Edit')).Text,1, i-1))='/ME') and
                                        (IRC_net_PageControl.ActivePage.Caption[1]='#') then
                                            begin
                                                IdIRC_NET.WriteLine('PRIVMSG
'+IRC_net_PageControl.ActivePage.Caption+' :[] ACTION
'+Copy(TEdit(FindComponent(IRC_net_PageControl.ActivePage.Name+'_Edit')).Text,i+1,Length(TEdit(FindComponent(IRC_net_PageControl.ActivePage.Name+'_Edit')).Text)-i)+'[] ');
                                            end;
                                TIRC_NETMemo(FindComponent(IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).AddFormattedString(TAG_COLOR+'6* '+IdIRC_NET.Nick+'
'+Copy(TEdit(FindComponent(IRC_net_PageControl.ActivePage.Name+'_Edit')).Text,i+1,Length(TEdit(FindComponent(IRC_net_PageControl.ActivePage.Name+'_Edit')).Text)-i));
                                // Робота з файлами лога
                                IRC_NETLogs(IRC_net_PageControl.ActivePage.Caption,IdIRC_NET.Nick,Copy(TEdit(FindComponent(IRC_net_PageControl.ActivePage.Name+'_Edit')).Text,i+1,Length(TEdit(FindComponent(IRC_net_PageControl.ActivePage.Name+'_Edit')).Text)-i));

```

```

TEdit (FindComponent (IRC_net_PageControl.ActivePage.Name+'_Edit')).Text:='';
    end

    else

        // QUIT
        if
AnsiUpperCase (Copy (TEdit (FindComponent (IRC_net_PageControl.ActivePage.Name+'_Edit')).Text,1, i-1))='/QUIT' then
            begin

IdIRC_NET.Quit (Copy (TEdit (FindComponent (IRC_net_PageControl.ActivePage.Name+'_Edit')).Text,i+1,Length (TEdit (FindComponent (IRC_net_PageControl.ActivePage.Name+'_Edit')).Text)-i)+' - '+IdIRC_NET.Replies.ClientInfo);
                IdIRC_NET.Disconnect;

TEdit (FindComponent (IRC_net_PageControl.ActivePage.Name+'_Edit')).Text:='';
                    end

                else

                    // AWAY
                    if
AnsiUpperCase (Copy (TEdit (FindComponent (IRC_net_PageControl.ActivePage.Name+'_Edit')).Text,1, i-1))='/AWAY' then
                        begin
                            IdIRC_NET.WriteLine ('AWAY
: '+Copy (TEdit (FindComponent (IRC_net_PageControl.ActivePage.Name+'_Edit')).Text,i+1,Length (TEdit (FindComponent (IRC_net_PageControl.ActivePage.Name+'_Edit')).Text)-i));

away_message:=Copy (TEdit (FindComponent (IRC_net_PageControl.ActivePage.Name+'_Edit')).Text,i+1,Length (TEdit (FindComponent (IRC_net_PageControl.ActivePage.Name+'_Edit')).Text)-i);

TEdit (FindComponent (IRC_net_PageControl.ActivePage.Name+'_Edit')).Text:='';
                            end

                            else

                                // MSG (Особисте повідомлення)
                                if
AnsiUpperCase (Copy (TEdit (FindComponent (IRC_net_PageControl.ActivePage.Name+'_Edit')).Text,1, i-1))='/MSG' then
                                    begin
                                        // Робота з файлами лога

IRC_NETLogs (IRC_net_PageControl.ActivePage.Caption,IdIRC_NET.Nick,Copy (TEdit (FindComponent (IRC_net_PageControl.ActivePage.Name+'_Edit')).Text,i+1,Length (TEdit (FindComponent (IRC_net_PageControl.ActivePage.Name+'_Edit')).Text)-i));

                                        // Якщо вкладка привата з такою назвою вже існує, то
відкриваємо її й відправляємо повідомлення
                                        for ip:=0 to IRC_net_PageControl.PageCount-1 do
                                            begin
                                                if IRC_net_PageControl.Pages[ip].Caption=Copy (s,0, c-1)
then
                                                    begin
                                                        // Відправляємо повідомлення
                                                        IdIRC_NET.WriteLine ('PRIVMSG '+Copy (s,0, c-1)+'
: '+Copy (s,c+1,Length (TEdit (FindComponent (IRC_net_PageControl.ActivePage.Name+'_Edit')).Text)-c+1));

TIRC_NETMemo (FindComponent (IRC_net_PageControl.Pages[ip].Name+'_IRC_NETMemo')).AddFormattedString (TAG_COLOR+'1<'+IdIRC_NET.Nick+'>
'+Copy (TEdit (FindComponent (IRC_net_PageControl.ActivePage.Name+'_Edit')).Text,i+

```

```

1,Length(TEdit(FindComponent(IRC_net_PageControl.ActivePage.Name+'_Edit')).Text)
-i));

TEdit(FindComponent(IRC_net_PageControl.ActivePage.Name+'_Edit')).Text:='';
// Перемикаємося на нову вкладку

IRC_net_PageControl.ActivePage:=IRC_net_PageControl.Pages[ip];
end;
end;

// Якщо вкладка привата з такою назвою вже існує, то зупиняємо
процедуру
for np:=0 to IRC_net_PageControl.PageCount-1 do
begin
if IRC_net_PageControl.Pages[np].Caption=Copy(s,0, c-1)
then
begin
// Зупиняємо процедуру
Exit;
end;
end;

// Якщо вкладки немає, то відправляємо повідомлення й
показуємо його текст в активному вікні
IdIRC_NET.WriteLine('PRIVMSG '+Copy(s,0, c-1)+'
:'+Copy(s,c+1,Length(TEdit(FindComponent(IRC_net_PageControl.ActivePage.Name+'_E
dit')).Text)-c+1));

TIRC_NETMemo(FindComponent(IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString(TAG_COLOR+'5<'+IdIRC_NET.Nick+'>
'+Copy(TEdit(FindComponent(IRC_net_PageControl.ActivePage.Name+'_Edit')).Text,i+
1,Length(TEdit(FindComponent(IRC_net_PageControl.ActivePage.Name+'_Edit')).Text)
-i));

TEdit(FindComponent(IRC_net_PageControl.ActivePage.Name+'_Edit')).Text:='';
end
else
// QUERY (Особисте повідомлення й створення вкладки для привата)
if
AnsiUpperCase(Copy(TEdit(FindComponent(IRC_net_PageControl.ActivePage.Name+'_E
dit')).Text,1, i-1))='/QUERY' then
begin
// Робота з файлами лога

IRC_NETLogs(IRC_net_PageControl.ActivePage.Caption,IdIRC_NET.Nick,Copy(TEdit(Fin
dComponent(IRC_net_PageControl.ActivePage.Name+'_Edit')).Text,i+1,Length(TEdit(F
indComponent(IRC_net_PageControl.ActivePage.Name+'_Edit')).Text)-i));

// Якщо вкладка привата з такою назвою вже існує, то
відкриваємо її й відправляємо повідомлення
for ip:=0 to IRC_net_PageControl.PageCount-1 do
begin
if IRC_net_PageControl.Pages[ip].Caption=Copy(s,0, c-1)
then
begin
// Відправляємо повідомлення
IdIRC_NET.WriteLine('PRIVMSG '+Copy(s,0, c-1)+'
:'+Copy(s,c+1,Length(TEdit(FindComponent(IRC_net_PageControl.ActivePage.Name+'_E
dit')).Text)-c+1));

TIRC_NETMemo(FindComponent(IRC_net_PageControl.Pages[ip].Name+'_IRC_NETMemo')).A
ddFormattedString(TAG_COLOR+'1<'+IdIRC_NET.Nick+'>
'+Copy(TEdit(FindComponent(IRC_net_PageControl.ActivePage.Name+'_Edit')).Text,i+
1,Length(TEdit(FindComponent(IRC_net_PageControl.ActivePage.Name+'_Edit')).Text)
-i));

TEdit(FindComponent(IRC_net_PageControl.ActivePage.Name+'_Edit')).Text:='';

```

```

// Перемикаємося на нову вкладку

IRC_net_PageControl.ActivePage:=IRC_net_PageControl.Pages[ip];
    end;
end;

// Якщо вкладка привата з такою назвою вже існує, то зупиняємо
процедуру
for np:=0 to IRC_net_PageControl.PageCount-1 do
begin
    if IRC_net_PageControl.Pages[np].Caption=Copy(s,0, c-1)
then
    begin
        // Зупиняємо процедуру
        Exit;
    end;
end;

// Якщо вкладки немає, те створюємо її нову вкладку й додаємо
в неї повідомлення
IRC_netNewPrivatPage(Copy(s,0, c-
1),IdIRC_NET.Nick,Copy(TEdit(FindComponent(IRC_net_PageControl.ActivePage.Name+'
_Edit')).Text,i+1,Length(TEdit(FindComponent(IRC_net_PageControl.ActivePage.Name+'
_Edit')).Text)-i),7);
    // Відправляємо повідомлення
    IdIRC_NET.WriteLine('PRIVMSG '+Copy(s,0, c-1)+'
:'+Copy(s,c+1,Length(TEdit(FindComponent(IRC_net_PageControl.ActivePage.Name+'_E
dit')).Text)-c+1));

TEdit(FindComponent(IRC_net_PageControl.ActivePage.Name+'_Edit')).Text:='';
    // Перемикаємося на нову вкладку

IRC_net_PageControl.ActivePageIndex:=IRC_net_PageControl.PageCount-1;
    end

    else

        // TOPIC
        if
AnsiUpperCase(Copy(TEdit(FindComponent(IRC_net_PageControl.ActivePage.Name+'_E
dit')).Text,1, i-1))='/TOPIC' then
        begin
            IdIRC_NET.WriteLine('TOPIC '+Copy(s,0, c-1)+'
:'+Copy(s,c+1,Length(TEdit(FindComponent(IRC_net_PageControl.ActivePage.Name+'_E
dit')).Text)-c+1));

TEdit(FindComponent(IRC_net_PageControl.ActivePage.Name+'_Edit')).Text:='';
            end

        else

            // NOTICE
            if
AnsiUpperCase(Copy(TEdit(FindComponent(IRC_net_PageControl.ActivePage.Name+'_E
dit')).Text,1, i-1))='/NOTICE' then
            begin
                IdIRC_NET.WriteLine('NOTICE '+Copy(s,0, c-1)+'
:'+Copy(s,c+1,Length(TEdit(FindComponent(IRC_net_PageControl.ActivePage.Name+'_E
dit')).Text)-c+1));

TIRC_NETMemo(FindComponent(IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString(TAG_COLOR+'7* - '+IdIRC_NET.Nick+' -
'+Copy(TEdit(FindComponent(IRC_net_PageControl.ActivePage.Name+'_Edit')).Text,i+
1,Length(TEdit(FindComponent(IRC_net_PageControl.ActivePage.Name+'_Edit')).Text)
-i));

TEdit(FindComponent(IRC_net_PageControl.ActivePage.Name+'_Edit')).Text:='';
            end

```

```

else

    // CLEAR - очищення вікна
    if
AnsiUpperCase (TEdit (FindComponent (IRC_net_PageControl.ActivePage.Name+'_Edit')).
Text)='/CLEAR' then
        begin

TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
Clear;

TEdit (FindComponent (IRC_net_PageControl.ActivePage.Name+'_Edit')).Text:='';
        end

    else

        // HELP
        if
AnsiUpperCase (TEdit (FindComponent (IRC_net_PageControl.ActivePage.Name+'_Edit')).
Text)='/HELP' then
            begin
                IRC_net_Faq_SpeedButtonClick(nil);

TEdit (FindComponent (IRC_net_PageControl.ActivePage.Name+'_Edit')).Text:='';
            end

        else

            // будь-яка інша команда
            begin

IdIRC_NET.WriteLine (Copy (TEdit (FindComponent (IRC_net_PageControl.ActivePage.Name+
'_Edit')).Text, 2, Length (TEdit (FindComponent (IRC_net_PageControl.ActivePage.Name+
'_Edit')).Text)));

TEdit (FindComponent (IRC_net_PageControl.ActivePage.Name+'_Edit')).Text:='';
            end;
        end

    else

        // повідомлення на канал і приват
        begin
            IdIRC_NET.WriteLine ('PRIVMSG
'+IRC_net_PageControl.ActivePage.Caption+
: '+TEdit (FindComponent (IRC_net_PageControl.ActivePage.Name+'_Edit')).Text);

TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString (TAG_COLOR+'1<'+IdIRC_NET.Nick+'>
'+TEdit (FindComponent (IRC_net_PageControl.ActivePage.Name+'_Edit')).Text);

            // Робота з файлами лога для каналу й привата

IRC_NETLogs (IRC_net_PageControl.ActivePage.Caption, IdIRC_NET.Nick, TEdit (FindComp
onent (IRC_net_PageControl.ActivePage.Name+'_Edit')).Text);

TEdit (FindComponent (IRC_net_PageControl.ActivePage.Name+'_Edit')).Text:='';
            end;
        except
            on e:Exception do Reconnect (E.Message);
        end;
    end;
end;

// Процедура вставки спецтегів у поле введення вікна сервера, каналу й привата
procedure TForm1.IRC_net_Channels_EditKeyDown (Sender: TObject;
var Key: Word; Shift: TShiftState);

```

```

var
  C : Char;
begin
if IRC_netColorForm.Showing=true then IRC_netColorForm.Close;
  C := UpCase(Char(Key));
  if (C in ['B','U','R','K',#13]) and (ssCtrl in Shift) then
    begin
      case C of
        'B' :
TEdit (FindComponent (IRC_net_PageControl.ActivePage.Name+'_Edit')).SelText:=TAG_B
OLD;
        'U' :
TEdit (FindComponent (IRC_net_PageControl.ActivePage.Name+'_Edit')).SelText:=TAG_U
NDERLINE;
        'R' :
TEdit (FindComponent (IRC_net_PageControl.ActivePage.Name+'_Edit')).SelText:=TAG_I
NVERSE;
        'K' : begin

IRC_netColorForm.Top:=Form1.Top+TIRC_NETMemo (FindComponent (IRC_net_PageControl.A
ctivePage.Name+'_IRC_NETMemo')).Height+90;
          IRC_netColorForm.Left:=Form1.Left+100;
          IRC_netColorForm.Show;

TEdit (FindComponent (IRC_net_PageControl.ActivePage.Name+'_Edit')).SelText :=
TAG_COLOR;
          end;
        #13 : begin
          if (IRC_net_PageControl.ActivePage.Caption[1]='#') and
(idIRC_NET.Connected) then
            begin
              // me (Action)
              try
                IdIRC_NET.WriteLine ('PRIVMSG
'+IRC_net_PageControl.ActivePage.Caption+' : ACTION
'+TEdit (FindComponent (IRC_net_PageControl.ActivePage.Name+'_Edit')).Text+' ');
              except
                on e:Exception do Reconnect (E.Message);
              end;
            end;

TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString (TAG_COLOR+'6* '+IdIRC_NET.Nick+
'+TEdit (FindComponent (IRC_net_PageControl.ActivePage.Name+'_Edit')).Text);

          // Робота з файлами логів

IRC_NETLogs (IRC_net_PageControl.ActivePage.Caption, IdIRC_NET.Nick, TEdit (FindComp
onent (IRC_net_PageControl.ActivePage.Name+'_Edit')).Text);

TEdit (FindComponent (IRC_net_PageControl.ActivePage.Name+'_Edit')).Text:='';
          end;
        end;
      end;
      Key := 0;
    end;
  end;

// Перехід на Edit при відпусканні миші
procedure TForm1.IRC_net_Server_IRC_NETMemoMouseUp (Sender: TObject;
  Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
begin
Form1.ActiveControl:=TEdit (FindComponent (IRC_net_PageControl.ActivePage.Name+'_E
dit'));

TEdit (FindComponent (IRC_net_PageControl.ActivePage.Name+'_Edit')).SelStart:=Leng
th (TEdit (FindComponent (IRC_net_PageControl.ActivePage.Name+'_Edit')).Text);
end;

```

```

// Процедура кліку за списком користувачів на каналі
procedure TForm1.IRC_net_Channels_ListBoxClick(Sender: TObject);
begin
    if
        TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex <> -1 then

        TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Hint:=TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex];
    end;

// Процедура показу кількості користувачів на каналу
procedure TForm1.IRC_net_Channel_ListBoxMouseMove(Sender: TObject;
    Shift: TShiftState; X, Y: Integer);
begin
    // Кількість користувачів у каналі

    TSplitter(FindComponent(IRC_net_PageControl.ActivePage.Name+'_Splitter')).Hint:=
        'Користувачів на каналі
        '+IntToStr(TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.Count);
    end;

// PopOp меню списку користувачів на каналі

// Додати нік у поле уведення
procedure TForm1.IRC_net_Channels_ListBox_PopUpMenu_AddNickClick(
    Sender: TObject);
begin
    if
        TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex <> -1 then
        begin
            // Якщо перший символ у ніку рівний + % @ те додаємо нік бех першого символу -
            інакше просто додаємо нік
            if
                (TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex][1]='+') or
                (TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex][1]='%') or
                (TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex][1]='@') then

                TEdit(FindComponent(IRC_net_PageControl.ActivePage.Name+'_Edit')).Text:=TEdit(FindComponent(IRC_net_PageControl.ActivePage.Name+'_Edit')).Text+Copy(TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex],2,Length(TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex]))+' '
            else

                TEdit(FindComponent(IRC_net_PageControl.ActivePage.Name+'_Edit')).Text:=TEdit(FindComponent(IRC_net_PageControl.ActivePage.Name+'_Edit')).Text+TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex]+' ';
            end
        else
            TIRC_NETMemo(FindComponent(IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).AddFormattedString(TAG_COLOR+'4Виберіть користувача зі списку');
        end;
end;

```

```

// Whois
procedure TForm1.IRC_net_Channels_ListBox_PopupMenu_WhoisClick(
  Sender: TObject);
begin
  if
    TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex > -1 then
    begin
      try
        if
          (TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex][1]='+') or
          (TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex][1]='%') or
          (TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex][1]='@') then
            IdIRC_NET.WriteLine('WHOIS
'+Copy(TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex],2,Length(TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex]))
            else
              IdIRC_NET.WriteLine('WHOIS
'+TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex]);
            except
              on e:Exception do Reconnect(E.Message);
            end;
          end
        else
          TIRC_NETMemo(FindComponent(IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).AddFormattedString(TAG_COLOR+'4Виберіть користувача зі списку');
        end;

// Приват
procedure TForm1.IRC_net_Channels_ListBox_PopupMenu_PrivatClick(
  Sender: TObject);
var
  i      : Integer;
  s      : String;
begin
  if
    TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex > -1 then
    begin
      // Одержуємо назву вкладки привата
      if
        (TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex][1]='+') or
        (TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex][1]='%') or
        (TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex][1]='@') then
          s:=Copy(TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListB

```

```

ox')).ItemIndex],2,Length(TListBox(FindComponent(IRC_net_PageControl.ActivePage.
Name+'_ListBox')).Items.Strings[TListBox(FindComponent(IRC_net_PageControl.Activ
ePage.Name+'_ListBox')).ItemIndex]))
    else

s:=TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items
.Strings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox'))
.ItemIndex];

    // Якщо вкладка привата з такою назвою вже існує, то відкриваємо її й
зупиняємо процедуру
    for i:=0 to IRC_net_PageControl.PageCount-1 do
        begin
            if IRC_net_PageControl.Pages[i].Caption=s then
                begin
                    // Перемикаємося на нову вкладку
                    IRC_net_PageControl.ActivePage:=IRC_net_PageControl.Pages[i];
                    // Зупиняємо процедуру
                    Exit;
                end;
            end;
        end;

    // Якщо вкладки немає, те створюємо її нову вкладку
    IRC_netNewPrivatPage(s, '', '', 7);

    // Перемикаємося на нову вкладку
    IRC_net_PageControl.ActivePageIndex:=IRC_net_PageControl.PageCount-1;
end
else

TIRC_NETMemo(FindComponent(IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString(TAG_COLOR+'4Виберіть користувача зі списку');
end;

// Керування . Оп
procedure TForm1.IRC_net_Channels_ListBox_PopupMenu_Control_OpClick(
    Sender: TObject);
begin
    if
TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemInde
x<>-1 then
        begin
            try
                if
(TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.S
trings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).I
temIndex][1]='+') or

(TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.S
trings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).I
temIndex][1]='%') or

(TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.S
trings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).I
temIndex][1]='@') then
                    IdIRC_NET.WriteLine('MODE '+IRC_net_PageControl.ActivePage.Caption+' +o
'+Copy(TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).I
tems.Strings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBo
x')).ItemIndex],2,Length(TListBox(FindComponent(IRC_net_PageControl.ActivePage.N
ame+'_ListBox')).Items.Strings[TListBox(FindComponent(IRC_net_PageControl.Active
Page.Name+'_ListBox')).ItemIndex]))
                else
                    IdIRC_NET.WriteLine('MODE '+IRC_net_PageControl.ActivePage.Caption+' +o
'+TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.
Strings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).
ItemIndex]);
            except
                on e:Exception do Reconnect(E.Message);
            end;
        end;
end;

```

```

end;
end
else
TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString (TAG_COLOR+'4Виберіть користувача зі списку');
end;

// Керування . Деоп
procedure TForm1.IRC_net_Channels_ListBox_PopupMenu_Control_DeOpClick(Sender:
TObject);
begin
if
TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemInde
x<>-1 then
begin
try
if
(TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.S
trings [TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).I
temIndex] [1]='+') or

(TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.S
trings [TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).I
temIndex] [1]='%') or

(TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.S
trings [TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).I
temIndex] [1]='@') then
    IdIRC_NET.WriteLine ('MODE '+IRC_net_PageControl.ActivePage.Caption+' -o
'+Copy (TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBo
x')).ItemIndex], 2, Length (TListBox (FindComponent (IRC_net_PageControl.ActivePage.N
ame+'_ListBox')).Items.Strings [TListBox (FindComponent (IRC_net_PageControl.Active
Page.Name+'_ListBox')).ItemIndex]))
    else
    IdIRC_NET.WriteLine ('MODE '+IRC_net_PageControl.ActivePage.Caption+' -o
'+TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.
Strings [TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).
ItemIndex]);
except
on e:Exception do Reconnect (E.Message);
end;
end
else
TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString (TAG_COLOR+'4Виберіть користувача зі списку');
end;

// Керування . Полуоп
procedure TForm1.IRC_net_Channels_ListBox_PopupMenu_Control_PoluOpClick(
Sender: TObject);
begin
if
TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemInde
x<>-1 then
begin
try
if
(TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.S
trings [TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).I
temIndex] [1]='+') or

(TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.S
trings [TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).I
temIndex] [1]='%') or

(TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.S
trings [TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).I
temIndex] [1]='@') then

```

```

        IdIRC_NET.WriteLine('MODE '+IRC_net_PageControl.ActivePage.Caption+' +h
'+Copy(TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).I
tems.Strings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBo
x')).ItemIndex],2,Length(TListBox(FindComponent(IRC_net_PageControl.ActivePage.N
ame+'_ListBox')).Items.Strings[TListBox(FindComponent(IRC_net_PageControl.Active
Page.Name+'_ListBox')).ItemIndex]))
        else
        IdIRC_NET.WriteLine('MODE '+IRC_net_PageControl.ActivePage.Caption+' +h
'+TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.
Strings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).
ItemIndex]);
    except
        on e:Exception do Reconnect(E.Message);
    end;
end
end
else
TIRC_NETMemo(FindComponent(IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString(TAG_COLOR+'4Виберіть користувача зі списку');
end;

// Керування . Забрати полуопа
procedure TForm1.IRC_net_Channels_ListBox_PopupMenu_Control_DePoluOpClick(
    Sender: TObject);
begin
    if
    TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemInde
x<>-1 then
    begin
        try
            if
            (TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.S
trings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).I
temIndex][1]='+') or

            (TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.S
trings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).I
temIndex][1]='%') or

            (TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.S
trings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).I
temIndex][1]='@') then
                IdIRC_NET.WriteLine('MODE '+IRC_net_PageControl.ActivePage.Caption+' -h
'+Copy(TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).I
tems.Strings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBo
x')).ItemIndex],2,Length(TListBox(FindComponent(IRC_net_PageControl.ActivePage.N
ame+'_ListBox')).Items.Strings[TListBox(FindComponent(IRC_net_PageControl.Active
Page.Name+'_ListBox')).ItemIndex]))
                else
                IdIRC_NET.WriteLine('MODE '+IRC_net_PageControl.ActivePage.Caption+' -h
'+TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.
Strings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).
ItemIndex]);
        except
            on e:Exception do Reconnect(E.Message);
        end;
    end
    else
    TIRC_NETMemo(FindComponent(IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString(TAG_COLOR+'4Виберіть користувача зі списку');
end;

// Керування . Войс
procedure TForm1.IRC_net_Channels_ListBox_PopupMenu_Control_VoiceClick(Sender:
TObject);
begin
    if
    TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemInde
x<>-1 then
    begin

```

```

try
  if
    (TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex][1]='+') or

    (TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex][1]='%') or

    (TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex][1]='@') then
      IdIRC_NET.WriteLine('MODE '+IRC_net_PageControl.ActivePage.Caption+' +v'+Copy(TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex],2,Length(TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex])))
    else
      IdIRC_NET.WriteLine('MODE '+IRC_net_PageControl.ActivePage.Caption+' +v'+TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex]);
  except
    on e:Exception do Reconnect(E.Message);
  end;
end;
end;
else
  TIRC_NETMemo(FindComponent(IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).AddFormattedString(TAG_COLOR+'4Виберіть користувача зі списку');
end;

// Керування . Девойс
procedure TForm1.IRC_net_Channels_ListBox_PopupMenu_Control_DeVoiceClick(Sender: TObject);
begin
  if
    TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex<>-1 then
    begin
      try
        if
          (TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex][1]='+') or

          (TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex][1]='%') or

          (TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex][1]='@') then
            IdIRC_NET.WriteLine('MODE '+IRC_net_PageControl.ActivePage.Caption+' -v'+Copy(TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex],2,Length(TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex])))
          else
            IdIRC_NET.WriteLine('MODE '+IRC_net_PageControl.ActivePage.Caption+' -v'+TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex]);
        except
          on e:Exception do Reconnect(E.Message);
        end;
      end;
    end;
  end;
end;

```

```

else
TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString (TAG_COLOR+'4Виберіть користувача зі списку');
end;

// Керування . Бан
procedure TForm1.IRC_net_Channels_ListBox_PopupMenu_Control_KickClick(Sender:
TObject);
begin
if
TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemInde
x<>-1 then
begin
try
if
(TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.S
trings [TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).I
temIndex] [1]='+') or

(TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.S
trings [TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).I
temIndex] [1]='%') or

(TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.S
trings [TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).I
temIndex] [1]='@') then
    IdIRC_NET.WriteLine ('KICK '+IRC_net_PageControl.ActivePage.Caption+
'+Copy (TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).I
tems.Strings [TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBo
x')).ItemIndex], 2, Length (TListBox (FindComponent (IRC_net_PageControl.ActivePage.N
ame+'_ListBox')).Items.Strings [TListBox (FindComponent (IRC_net_PageControl.Active
Page.Name+'_ListBox')).ItemIndex]))
    else
    IdIRC_NET.WriteLine ('KICK '+IRC_net_PageControl.ActivePage.Caption+
'+TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.
Strings [TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).
ItemIndex]);
except
on e:Exception do Reconnect (E.Message);
end;
end
else
TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString (TAG_COLOR+'4Виберіть користувача зі списку');
end;

// Бан
procedure TForm1.IRC_net_Channels_ListBox_PopupMenu_Control_BanClick(
Sender: TObject);
begin
if
TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemInde
x<>-1 then
begin
try
if
(TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.S
trings [TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).I
temIndex] [1]='+') or

(TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.S
trings [TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).I
temIndex] [1]='%') or

(TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.S
trings [TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).I
temIndex] [1]='@') then
    IdIRC_NET.WriteLine ('MODE '+IRC_net_PageControl.ActivePage.Caption+ ' +b
'+Copy (TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).I

```

```

tems.Strings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBo
x')).ItemIndex],2,Length(TListBox(FindComponent(IRC_net_PageControl.ActivePage.N
ame+'_ListBox')).Items.Strings[TListBox(FindComponent(IRC_net_PageControl.Active
Page.Name+'_ListBox')).ItemIndex]))
    else
        IdIRC_NET.WriteLine('MODE '+IRC_net_PageControl.ActivePage.Caption+' +b
'+TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.
Strings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).
ItemIndex]);
except
    on e:Exception do Reconnect(E.Message);
end;
end
else
TIRC_NETMemo(FindComponent(IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString(TAG_COLOR+'4Виберіть користувача зі списку');
end;

// CTCP запити
procedure TForm1.IRC_net_Channels_ListBox_PopupMenu_CTCP_Click(
    Sender: TObject);
var
    s : String;
begin
with sender as TMenuItem do s:=Caption;
if
TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemInde
x<>-1 then
begin
try
    if
(TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.S
trings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).I
temIndex][1]='+') or

(TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.S
trings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).I
temIndex][1]='%') or

(TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.S
trings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).I
temIndex][1]='@') then
        IdIRC_NET.WriteLine('PRIVMSG
'+Copy(TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBo
x')).ItemIndex],2,Length(TListBox(FindComponent(IRC_net_PageControl.ActivePage.N
ame+'_ListBox')).Items.Strings[TListBox(FindComponent(IRC_net_PageControl.Active
Page.Name+'_ListBox')).ItemIndex]))+' :| '+AnsiUpperCase(s)+'| ')
    else
        IdIRC_NET.WriteLine('PRIVMSG
'+TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.
Strings[TListBox(FindComponent(IRC_net_PageControl.ActivePage.Name+'_ListBo
x')).ItemIndex]+' :| '+AnsiUpperCase(s)+'| ');
except
    on e:Exception do Reconnect(E.Message);
end;
end
else
TIRC_NETMemo(FindComponent(IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString(TAG_COLOR+'4Виберіть користувача зі списку');
end;

// Action
procedure TForm1.IRC_net_Channels_ListBox_PopupMenu_Action_Click(Sender:
TObject);
var
    s : String;
begin
with sender as TMenuItem do s:=Caption;

```

```

if
TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex<-1 then
begin
try
if
(TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex][1]='+') or

(TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex][1]='%') or

(TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex][1]='@') then
begin
IdIRC_NET.WriteLine ('PRIVMSG '+IRC_net_PageControl.ActivePage.Caption+' :[]
ACTION '+s+'
'+Copy (TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex],2,Length (TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex]))+'[] ');

TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).AddFormattedString (TAG_COLOR+'6* '+IdIRC_NET.Nick+' '+s+'
'+Copy (TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex],2,Length (TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex])));
// Робота з файлами лога
IRC_NETLogs (IRC_net_PageControl.ActivePage.Caption,IdIRC_NET.Nick,s+'
'+Copy (TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex],2,Length (TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex])));
end
else
begin
IdIRC_NET.WriteLine ('PRIVMSG '+IRC_net_PageControl.ActivePage.Caption+' :[]
ACTION '+s+'
'+TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex]+'[] ');

TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).AddFormattedString (TAG_COLOR+'6* '+IdIRC_NET.Nick+' '+s+'
'+TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex]);
// Робота з файлами лога
IRC_NETLogs (IRC_net_PageControl.ActivePage.Caption,IdIRC_NET.Nick,s+'
'+TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox (FindComponent (IRC_net_PageControl.ActivePage.Name+'_ListBox')).ItemIndex]);
end;
except
on e:Exception do Reconnect (E.Message);
end;
end
else
TIRC_NETMemo (FindComponent (IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).AddFormattedString (TAG_COLOR+'4Виберіть користувача зі списку');
end;

```

```

// PopUp меню списку каналів
// Увійти на канал
procedure TForm1.Go_To_ChannelClick(Sender: TObject);
var
  i : Integer;
begin
  if Channels_ListBox.ItemIndex=-1 then exit;
  i:=Pos(' ',Channels_ListBox.Items.Strings[Channels_ListBox.ItemIndex]);
  try

    IdIRC_NET.Join(Copy(Channels_ListBox.Items.Strings[Channels_ListBox.ItemIndex],1
    , i-1));
  except
    // Помилка
    on e:Exception do
      TIRC_NETMemo(FindComponent(IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
      AddFormattedString(TAG_COLOR+'4'+E.Message);
  end;
end;

// Клік за списком каналів
procedure TForm1.Channels_ListBoxClick(Sender: TObject);
begin
  if Channels_ListBox.ItemIndex<>-1 then

    Channels_ListBox.Hint:=ParseColorString(Channels_ListBox.Items.Strings[Channels_
    ListBox.ItemIndex]);
  end;

  // DCC
  // Підтвердження докачки файлу
  procedure TForm1.IdIRC_NETDCCAccept(Sender: TObject; ANick, APort, AFileName,
  APosition: String);
  begin

    TIRC_NETMemo(FindComponent(IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
    AddFormattedString(TAG_COLOR+'1IdIRC_NETDCCAccept '+'ANick='+ANick+'
    APort='+APort+' AFileName='+AFileName+' APosition='+APosition);
  end;

  // Користувач хоче поспілкуватися з Вами
  procedure TForm1.IdIRC_NETDCCChat(Sender: TObject; ANick, AIP, APort: String);
  var
    // Переводимо IP з DWORD у нормальний вид
    ip1,ip2,ip3,ip4 : Integer;
    ip : String;
  begin
    // Переводимо IP з DWORD у нормальний вид
    ip4:=StrToInt(AIP) div 16777216;
    ip3:=StrToInt(AIP) mod 16777216 div 65536;
    ip2:=StrToInt(AIP) mod 16777216 mod 65536 div 256;
    ip1:=StrToInt(AIP) mod 16777216 mod 65536 mod 256;
    ip:=IntToStr(ip4)+'.'+IntToStr(ip3)+'.'+IntToStr(ip2)+'.'+IntToStr(ip1);

    IRC_netDCCForm.Caption:='DCC чат';
    IRC_netDCCForm.IRC_netDCC_Label.Caption:='Користувач хоче поспілкуватися з
    вами через DCC чат';
    IRC_netDCCForm.IRC_netDCC_User_Label_.Caption:=ANick;
    IRC_netDCCForm.IRC_netDCC_IP_Label_.Caption:=ip;
    IRC_netDCCForm.IRC_netDCC_IP_Label_.Hint:=AIP;
    IRC_netDCCForm.IRC_netDCC_Port_Label_.Caption:=APort;
    IRC_netDCCForm.IRC_netDCC_FileName_Label.Visible:=false;
    IRC_netDCCForm.IRC_netDCC_FileName_Label_.Visible:=false;
    IRC_netDCCForm.IRC_netDCC_FileSize_Label.Visible:=false;
    IRC_netDCCForm.IRC_netDCC_FileSize_Label_.Visible:=false;
    IRC_netDCCForm.IRC_netDCC_FileSize_Label_.Visible:=false;
    IRC_netDCCForm.IRC_netDCC_Gauge.Visible:=false;
    IRC_netDCCForm.IRC_netDCC_OK_Button.Top:=80;
  end;
end;

```

```

IRC_netDCCForm.IRC_netDCC_Close_Button.Top:=80;
IRC_netDCCForm.Height:=163;
IRC_netDCCForm.IRC_netDCC_OK_Button.Enabled:=true;
IRC_netDCCForm.IRC_netDCC_Close_Button.Caption:='Скасування';
IRC_netDCCForm.IRC_netDCC_StatusBar.Panels[0].Text:='';
IRC_netDCCForm.Show;
end;

// Запит на докачку файлу
procedure TForm1.IdIRC_NETDCCResume(Sender: TObject; ANick, APort, AFileName,
  APosition: String);
begin
TIRC_NETMemo (FindComponent(IRC_net_PageControl.ActivePage.Name+'_IRC_NETMemo')).
AddFormattedString(TAG_COLOR+'1IdIRC_NETDCCResume '+'ANick='+ANick+'
APort='+APort+' AFileName='+AFileName+' APosition='+APosition);
end;

// Користувач намагається послати вам файл
// Розмір файлу рядком
function BytesToStr(const i64Size: Int64): string;
const
  i64GB = 1024 * 1024 * 1024;
  i64MB = 1024 * 1024;
  i64KB = 1024;
begin
  if i64Size div i64GB > 0 then
    Result := Format('%.2f GB', [i64Size / i64GB])
  else if i64Size div i64MB > 0 then
    Result := Format('%.2f MB', [i64Size / i64MB])
  else if i64Size div i64KB > 0 then
    Result := Format('%.2f KB', [i64Size / i64KB])
  else
    Result := IntToStr(i64Size) + ' Byte(s)';
end;

procedure TForm1.IdIRC_NETDCCSend(Sender: TObject; ANick, AIP, APort,
  AFileName, AFileSize: String);
var
  // Переводимо IP з DWORD у нормальний вид
  ip1,ip2,ip3,ip4 : Integer;
  ip : String;
begin
  // Переводимо IP з DWORD у нормальний вид
  ip4:=StrToInt(AIP) div 16777216;
  ip3:=StrToInt(AIP) mod 16777216 div 65536;
  ip2:=StrToInt(AIP) mod 16777216 mod 65536 div 256;
  ip1:=StrToInt(AIP) mod 16777216 mod 65536 mod 256;
  ip:=IntToStr(ip4)+'.'+IntToStr(ip3)+'.'+IntToStr(ip2)+'.'+IntToStr(ip1);

  IRC_netDCCForm.Caption:='DCC прийом файлу';
  IRC_netDCCForm.IRC_netDCC_Label.Caption:='Користувач хоче відправити вам файл
через DCC';
  IRC_netDCCForm.IRC_netDCC_User_Label.Caption:=ANick;
  IRC_netDCCForm.IRC_netDCC_IP_Label.Caption:=ip;
  IRC_netDCCForm.IRC_netDCC_IP_Label.Hint:=AIP;
  IRC_netDCCForm.IRC_netDCC_Port_Label.Caption:=APort;
  IRC_netDCCForm.IRC_netDCC_FileName_Label.Caption:=AFileName;
  IRC_netDCCForm.IRC_netDCC_FileName_Label.Hint:=AFileName;
  IRC_netDCCForm.IRC_netDCC_FileSize_Label.Caption:=AFileSize;

  IRC_netDCCForm.IRC_netDCC_FileSizeLabel.Left:=136+IRC_netDCCForm.IRC_netDCC_File
Size_Label.Width+10;
  IRC_netDCCForm.IRC_netDCC_FileSizeLabel.Caption:='(
'+BytesToStr(StrToInt(AFileSize))+' )';
  IRC_netDCCForm.IRC_netDCC_FileName_Label.Visible:=true;
  IRC_netDCCForm.IRC_netDCC_FileName_Label.Visible:=true;
  IRC_netDCCForm.IRC_netDCC_FileSize_Label.Visible:=true;
  IRC_netDCCForm.IRC_netDCC_FileSizeLabel.Visible:=true;

```

```

IRC_netDCCForm.IRC_netDCC_FileSize_Label_.Visible:=true;
IRC_netDCCForm.IRC_netDCC_Gauge.Visible:=true;
IRC_netDCCForm.IRC_netDCC_OK_Button.Top:=152;
IRC_netDCCForm.IRC_netDCC_Close_Button.Top:=152;
IRC_netDCCForm.Height:=237;
IRC_netDCCForm.IRC_netDCC_Gauge.Progress:=0;
IRC_netDCCForm.IRC_netDCC_OK_Button.Enabled:=true;
IRC_netDCCForm.IRC_netDCC_Close_Button.Caption:='Скасування';
IRC_netDCCForm.IRC_netDCC_StatusBar.Panels[0].Text:='';
IRC_netDCCForm.Show;
end;

// Команди
// Включити режим AWAY
procedure TForm1.IRC_net_Commands_PopupMenu_Away_OnClick(Sender: TObject);
begin
IRC_netCommandsForm.Caption:='Включити режим AWAY';
IRC_netCommandsForm.IRC_net_Commands_GroupBox.Caption:='Уведіть текст
повідомлення';
IRC_netCommandsForm.IRC_net_Commands_Edit.Text:='';
IRC_netCommandsForm.ShowModal;
end;

// Виключити режим AWAY
procedure TForm1.IRC_net_Commands_PopupMenu_Away_OffClick(Sender: TObject);
begin
try
IdIRC_NET.WriteLine('AWAY');
except
on e:Exception do Reconnect(E.Message);
end;
end;

// Зайти в канал
procedure TForm1.IRC_net_Commands_PopupMenu_JoinClick(Sender: TObject);
begin
IRC_netCommandsForm.Caption:='Зайти в канал';
IRC_netCommandsForm.IRC_net_Commands_GroupBox.Caption:='Уведіть ім'я каналу (і
пароль якщо необхідно)';
IRC_netCommandsForm.IRC_net_Commands_Edit.Text:='#';
IRC_netCommandsForm.ShowModal;
end;

// Покинути канал
procedure TForm1.IRC_net_Commands_PopupMenu_PartClick(Sender: TObject);
begin
IRC_netCommandsForm.Caption:='Покинути канал';
IRC_netCommandsForm.IRC_net_Commands_GroupBox.Caption:='Уведіть ім'я каналу';
if IRC_net_PageControl.ActivePage.Caption[1]='#' then
IRC_netCommandsForm.IRC_net_Commands_Edit.Text:=IRC_net_PageControl.ActivePage.C
aption
else
IRC_netCommandsForm.IRC_net_Commands_Edit.Text:='#';
IRC_netCommandsForm.ShowModal;
end;

// Послати приват
procedure TForm1.IRC_net_Commands_PopupMenu_MsgClick(Sender: TObject);
begin
IRC_netCommandsForm.Caption:='Послати приват';
IRC_netCommandsForm.IRC_net_Commands_GroupBox.Caption:='Уведіть нік
користувача й текст повідомлення';
IRC_netCommandsForm.IRC_net_Commands_Edit.Text:='';
IRC_netCommandsForm.ShowModal;
end;

procedure TForm1.IRC_net_Commands_PopupMenu_NoticeClick(Sender: TObject);
begin

```

```

    IRC_netCommandsForm.Caption:='Послати Notice';
    IRC_netCommandsForm.IRC_net_Commands_GroupBox.Caption:='Уведіть нік користувача або ім'я каналу й текст повідомлення';
    IRC_netCommandsForm.IRC_net_Commands_Edit.Text:='';
    IRC_netCommandsForm.ShowModal;
end;

// Послати повідомлення від третьої особи
procedure TForm1.IRC_net_Commands_PopupMenu_MeClick(Sender: TObject);
begin
    IRC_netCommandsForm.Caption:='Послати Action';
    IRC_netCommandsForm.IRC_net_Commands_GroupBox.Caption:='Уведіть ім'я каналу й текст повідомлення';
    if IRC_net_PageControl.ActivePage.Caption[1]='#' then

IRC_netCommandsForm.IRC_net_Commands_Edit.Text:=IRC_net_PageControl.ActivePage.C
aption+' '
    else
    IRC_netCommandsForm.IRC_net_Commands_Edit.Text:='#';
    IRC_netCommandsForm.ShowModal;
end;

// Whois користувача (online)
procedure TForm1.IRC_net_Commands_PopupMenu_WhoisClick(Sender: TObject);
begin
    IRC_netCommandsForm.Caption:='Whois користувача (online)';
    IRC_netCommandsForm.IRC_net_Commands_GroupBox.Caption:='Уведіть нік користувача';
    IRC_netCommandsForm.IRC_net_Commands_Edit.Text:='';
    IRC_netCommandsForm.ShowModal;
end;

// Whowas користувача (offline)
procedure TForm1.IRC_net_Commands_PopupMenu_WhowasClick(Sender: TObject);
begin
    IRC_netCommandsForm.Caption:='Whowas користувача (offline)';
    IRC_netCommandsForm.IRC_net_Commands_GroupBox.Caption:='Уведіть нік користувача';
    IRC_netCommandsForm.IRC_net_Commands_Edit.Text:='';
    IRC_netCommandsForm.ShowModal;
end;

// Перемінити нік
procedure TForm1.IRC_net_Commands_PopupMenu_NickClick(Sender: TObject);
begin
    IRC_netCommandsForm.Caption:='Перемінити нік';
    IRC_netCommandsForm.IRC_net_Commands_GroupBox.Caption:='Уведіть новий нік';
    IRC_netCommandsForm.IRC_net_Commands_Edit.Text:='';
    IRC_netCommandsForm.ShowModal;
end;

// Послати STCP запити
procedure TForm1.IRC_net_Commands_PopupMenu_STCP_Click(Sender: TObject);
var
    s : String;
begin
    with sender as TMenuItem do s:=Caption;
    IRC_netCommandsForm.Caption:='Послати STCP запит '+s;
    IRC_netCommandsForm.IRC_net_Commands_GroupBox.Caption:='Уведіть нік користувача або ім'я каналу';
    IRC_netCommandsForm.IRC_net_Commands_Edit.Text:='';
    IRC_netCommandsForm.ShowModal;
end;

// Одержати інформацію про канал
procedure TForm1.IRC_net_Commands_PopupMenu_ChannelInfoClick(Sender: TObject);
begin
    IRC_netCommandsForm.Caption:='Одержати інформацію про канал';
    IRC_netCommandsForm.IRC_net_Commands_GroupBox.Caption:='Уведіть ім'я каналу';

```

```

if IRC_net_PageControl.ActivePage.Caption[1]='#' then

IRC_netCommandsForm.IRC_net_Commands_Edit.Text:=IRC_net_PageControl.ActivePage.C
aption
else
IRC_netCommandsForm.IRC_net_Commands_Edit.Text:='#';
IRC_netCommandsForm.ShowModal;
end;

// Одержати список серверів мережі
procedure TForm1.IRC_net_Commands_PopupMenu_LinksClick(Sender: TObject);
begin
try
IdIRC_NET.WriteLine('LINKS');
except
on e:Exception do Reconnect(E.Message);
end;
end;

// Одержати дані админа сервера
procedure TForm1.IRC_net_Commands_PopupMenu_AdminClick(Sender: TObject);
begin
try
IdIRC_NET.WriteLine('ADMIN');
except
on e:Exception do Reconnect(E.Message);
end;
end;

// Зареєструвати нік
procedure TForm1.IRC_net_Commands_PopupMenu_RegNickClick(Sender: TObject);
begin
IRC_netCommandsForm.Caption:='Зареєструвати нік';
IRC_netCommandsForm.IRC_net_Commands_GroupBox.Caption:='Уведіть пароль і E-
mail';
IRC_netCommandsForm.IRC_net_Commands_Edit.Text:='';
IRC_netCommandsForm.ShowModal;
end;

// Зареєструвати канал
procedure TForm1.IRC_net_Commands_PopupMenu_RegChannelClick(Sender: TObject);
begin
IRC_netCommandsForm.Caption:='Зареєструвати канал';
IRC_netCommandsForm.IRC_net_Commands_GroupBox.Caption:='Уведіть ім'я каналу,
пароль і опис каналу';
if IRC_net_PageControl.ActivePage.Caption[1]='#' then

IRC_netCommandsForm.IRC_net_Commands_Edit.Text:=IRC_net_PageControl.ActivePage.C
aption+' '
else
IRC_netCommandsForm.IRC_net_Commands_Edit.Text:='#';
IRC_netCommandsForm.ShowModal;
end;

// Перевірка пінга кожні 200 сек
procedure TForm1.IRC_net_Ping_TimerTimer(Sender: TObject);
begin
if IdIRC_NET.Connected then Reconnect('Перепідключення');
end;
////////////////////
// Кольоровий список каналів
function TForm1.CheckItemSize(Index: Integer): Integer;
procedure Toggle(FontStyle: TFontStyle);
begin
if FontStyle in Channels_ListBox.Canvas.Font.Style
then begin
Channels_ListBox.Canvas.Font.Style := Channels_ListBox.Canvas.Font.Style -
[FontStyle];
end else begin

```

```

    Channels_ListBox.Canvas.Font.Style := Channels_ListBox.Canvas.Font.Style +
[FontStyle];
    end;
end;
var
    x: Integer;
    Phrase: String;
    Current: Integer;
    NextSpace: Integer;
    LastSpace: Integer;
    WidthMax: Integer;
    Width: Integer;
    Size: Integer;
begin
    Result := 17;
    x := 3;
    Channels_ListBox.Canvas.Font := Channels_ListBox.Font;
    Phrase := Channels_ListBox.Items[Index];
    Current := 1;
    NextSpace := -1;
    LastSpace := -1;
    WidthMax := Channels_ListBox.ClientWidth - GetSystemMetrics(SM_CXVSCROLL);
    Width := x;
    Size := Length(Phrase);
    while Current <= Size
    do begin
        case Phrase[Current] of
            #02:
                begin
                    Toggle(fsBold);
                    Inc(Current);
                    Continue;
                end;
            #03:
                begin
                    if Phrase[Current + 1] in ['0'..'9']
                    then begin
                        if Phrase[Current + 2] in ['0'..'9']
                        then begin
                            Inc(Current, 3);
                        end else begin
                            Inc(Current, 2);
                        end;
                    end else begin
                        Inc(Current);
                    end;
                    if Phrase[Current] = ','
                    then begin
                        if Phrase[Current + 1] in ['0'..'9']
                        then begin
                            if Phrase[Current + 2] in ['0'..'9']
                            then begin
                                Inc(Current, 3);
                            end else begin
                                Inc(Current, 2);
                            end;
                        end else begin
                            Inc(Current);
                        end;
                    end;
                    Continue;
                end;
            #15:
                begin
                    Channels_ListBox.Canvas.Font := Channels_ListBox.Font;
                    Inc(Current);
                    Continue;
                end;
            #22:
                begin

```

```

        Toggle(fsItalic);
        Inc(Current);
        Continue;
    end;
    #31:
    begin
        Toggle(fsUnderline);
        Inc(Current);
        Continue;
    end;
    #32:
    begin
        NextSpace := Current;
    end;
end;
Inc(Width, Channels_ListBox.Canvas.TextWidth(Phrase[Current]));
if Width > WidthMax
then begin
    if LastSpace > 0
    then begin
        Current := LastSpace;
    end else begin
        Dec(Current);
    end;
    NextSpace := -1;
    Inc(Result, 17);
    x := 3;
    Width := x;
end;
LastSpace := NextSpace;
Inc(Current);
end;
end;

procedure TForm1.Channels_ListBoxMeasureItem(Control: TWinControl;
    Index: Integer; var Height: Integer);
begin
    Height := CheckItemSize(Index);
end;

procedure TForm1.Channels_ListBoxDrawItem(Control: TWinControl;
    Index: Integer; Rect: TRect; State: TOwnerDrawState);
procedure Toggle(FontStyle: TFontStyle);
begin
    if FontStyle in Channels_ListBox.Canvas.Font.Style
    then begin
        Channels_ListBox.Canvas.Font.Style := Channels_ListBox.Canvas.Font.Style -
[FontStyle];
    end else begin
        Channels_ListBox.Canvas.Font.Style := Channels_ListBox.Canvas.Font.Style +
[FontStyle];
    end;
end;
end;
var
    y: Integer;
    x: Integer;
    DefaultColor: TColor;
    Phrase: String;
    Current: Integer;
    NextSpace: Integer;
    LastSpace: Integer;
    LastPrint: Integer;
    WidthMax: Integer;
    Width: Integer;
    Size: Integer;
procedure Flush(Skip: Integer = 0);
var
    Count: Integer;
    Text: String;
begin

```

```

Count := Current - LastPrint + 1 - Skip;
if Count > 0
then begin
  Text := Copy(Phrase, LastPrint, Count);
  Channels_ListBox.Canvas.TextOut(x, y, Text);
  Inc(x, Channels_ListBox.Canvas.TextWidth(Text));
  Inc(LastPrint, Count);
end;
end;
begin
  Size := CheckItemSize(Index);
  if Size <> Channels_ListBox.Perform(LB_GETITEMHEIGHT, Index, 0)
  then begin
    Channels_ListBox.Perform(LB_SETITEMHEIGHT, Index, Size);
    Channels_ListBox.Invalidate;
    Exit;
  end;
  y := Rect.Top + 2;
  x := Rect.Left + 3;
  if odSelected in State
  then begin
    DefaultColor := $00FFCECE;
  end else begin
    DefaultColor :=Channels_ListBox.Color;
  end;
  Channels_ListBox.Canvas.Font :=Channels_ListBox.Font;
  Channels_ListBox.Canvas.Brush.Color := DefaultColor;
  Channels_ListBox.Canvas.FillRect(Rect);
  Phrase :=Channels_ListBox.Items[Index];
  Current := 1;
  NextSpace := -1;
  LastSpace := -1;
  LastPrint := 1;
  WidthMax :=Channels_ListBox.ClientWidth - GetSystemMetrics(SM_CXVSCROLL);
  Width := x;
  Size := Length(Phrase);
  while Current <= Size
  do begin
    case Phrase[Current] of
      #02:
        begin
          Flush(1);
          Toggle(fsBold);
          Inc(Current);
          LastPrint := Current;
          Continue;
        end;
      #03:
        begin
          Flush(1);
          if Phrase[Current + 1] in ['0'..'9']
          then begin
            if Phrase[Current + 2] in ['0'..'9']
            then begin
              Channels_ListBox.Canvas.Font.Color := Colors[StrToInt(Copy(Phrase,
Current + 1, 2)) mod 16];
              Inc(Current, 3);
              LastPrint := Current;
            end else begin
              Channels_ListBox.Canvas.Font.Color := Colors[StrToInt(Phrase[Current
+ 1]) mod 16];
              Inc(Current, 2);
              LastPrint := Current;
            end;
          end else begin
            Channels_ListBox.Canvas.Font.Color :=Channels_ListBox.Font.Color;
            Channels_ListBox.Canvas.Brush.Color := DefaultColor;
            Inc(Current);
            LastPrint := Current;
          end;
        end;
    end;
  end;
end;

```

```

end;
if Phrase[Current] = ','
then begin
  if Phrase[Current + 1] in ['0'..'9']
  then begin
    if Phrase[Current + 2] in ['0'..'9']
    then begin
      Channels_ListBox.Canvas.Brush.Color :=
Colors[StrToInt(Copy(Phrase, Current + 1, 2)) mod 16];
      Inc(Current, 3);
      LastPrint := Current;
    end else begin
      Channels_ListBox.Canvas.Brush.Color :=
Colors[StrToInt(Phrase[Current + 1]) mod 16];
      Inc(Current, 2);
      LastPrint := Current;
    end;
  end else begin
    Channels_ListBox.Canvas.Brush.Color := DefaultColor;
    Inc(Current);
    LastPrint := Current;
  end;
end;
end;
Continue;
end;
#15:
begin
  Flush(1);
  Channels_ListBox.Canvas.Font :=Channels_ListBox.Font;
  Channels_ListBox.Canvas.Brush.Color := DefaultColor;
  Inc(Current);
  LastPrint := Current;
  Continue;
end;
#22:
begin
  Flush(1);
  Toggle(fsItalic);
  Inc(Current);
  LastPrint := Current;
  Continue;
end;
#31:
begin
  Flush(1);
  Toggle(fsUnderline);
  Inc(Current);
  LastPrint := Current;
  Continue;
end;
#32:
begin
  NextSpace := Current;
end;
end;
Inc(Width, Channels_ListBox.Canvas.TextWidth(Phrase[Current]));
if Width > WidthMax
then begin
  if LastSpace > 0
  then begin
    Current := LastSpace;
  end else begin
    Dec(Current);
  end;
end;
Flush;
NextSpace := -1;
Inc(y, Channels_ListBox.ItemHeight);
x := Rect.Left + 3;
Width := x;

```

```

    end;
    LastSpace := NextSpace;
    Inc(Current);
end;
Flush;
if odFocused in State
then begin
    Channels_ListBox.Canvas.DrawFocusRect(Rect);
end;
end;
//////////
// Дії при відкритті програми
procedure TForm1.FormCreate(Sender: TObject);
var
    // Читання настроювань із ini файлу
    Ini          : Tinifile;
    // Розшифровка пароля IRC_NET
    s            : String[255];
    c            : array[0..255] of Byte absolute s;
    i            : Integer;
                // Говорилка
    NumFound    : DWord;
    ModeInfo    : TTSModeInfo;
begin
    ////////////
    // Заносимо дані у вікно сервера IRC_NET
    IRC_net_Server_IRC_NETMemo.AddFormattedString('
8,8
0,0 /join #имя_канала - Зайти
в канал');
    IRC_net_Server_IRC_NETMemo.AddFormattedString('
8,8
0,0
1,1
0,0
1,1
0,0
1,1
0,0
1,1
0,0
8,8
0,0 /nick ваш_новий_нік - Перемінити нік');
    IRC_net_Server_IRC_NETMemo.AddFormattedString('
8,8
0,0
4,4
1,1
0,0
4,4
1,1
0,0
4,4
1,1
0,0
4,4
1,1
0,0
8,8
0,0 /list - Показати список каналів');
    IRC_net_Server_IRC_NETMemo.AddFormattedString('
8,8
0,0
4,4
1,1
4,4
1,1
0,0
4,4
1,1
0,0
4,4
1,1
0,0
4,4
1,1
0,0
8,8
0,0 /msg нік_користувача текст - Відправити
особисте повідомлення');
    IRC_net_Server_IRC_NETMemo.AddFormattedString('
8,8
0,0
4,4
1,1
0,0
4,4
0,0
4,4
1,1
0,0
4,4
0,0
8,8
0,0
6/me текст або наберіть текст і натисніть Ctrl+Enter -');
    IRC_net_Server_IRC_NETMemo.AddFormattedString('
8,8
0,0
4,4
1,1
0,0
4,4
1,1
0,0
4,4
1,1
0,0
4,4
1,1
0,0
8,8
0,0 6Відправити Action на канал');
    IRC_net_Server_IRC_NETMemo.AddFormattedString('
8,8
0,0
4,4
0,0
4,4
0,0
4,4
0,0
4,4
0,0
8,8
0,0
7/notice нік_користувач або #имя_канала и текст -');
    IRC_net_Server_IRC_NETMemo.AddFormattedString('
8,8
0,0
7Відправити Notice');

    IRC_net_Server_IRC_NETMemo.AddFormattedString('/away текст - Включити
автовідповідь на особисті повідомлення (/away - Виключити)');
    IRC_net_Server_IRC_NETMemo.AddFormattedString('/clear - Очистити вікно показу
повідомлень');
    IRC_net_Server_IRC_NETMemo.AddFormattedString('/help - Показати довідник по
IRC_NET командах');
    IRC_net_Server_IRC_NETMemo.AddFormattedString('Ctrl+B - Жирний текст');
    IRC_net_Server_IRC_NETMemo.AddFormattedString('Ctrl+U - Підкреслений текст');
    IRC_net_Server_IRC_NETMemo.AddFormattedString('Ctrl+R - Інвертований текст');
    IRC_net_Server_IRC_NETMemo.AddFormattedString('Ctrl+K+номер кольору - 1Ц 2В
3е 4т 5н 6про 7й текст');
    IRC_net_Server_IRC_NETMemo.AddFormattedString('Ctrl+K+номер кольору+,+номер
кольору тла - 1Ц 2В 3е 4т 5н 6про 7й текст на кольоровому тлі');

    // Говорилка
    try
        // Ініціалізація аудіопристрою

```

```

CoCreateInstance(CLSID_MMAudioDest, Nil, CLSCTX_ALL,
IID_IAudioMultiMediaDevice, fIAMM);

// Створення об'єкта, що перераховується, для перебору всіх движків у
системі
// за допомогою інтерфейсу ITTSEnum
CoCreateInstance(CLSID_TTSEnumerator, Nil, CLSCTX_ALL, IID_ITTSEnum,
aTTSEnum);
// Скидаємо на перший
aTTSEnum.Reset;
// Одержуємо перший движок
aTTSEnum.Next(1, ModeInfo, @NumFound);
While NumFound > 0 do
begin
Speak_Combobox.Items.Add(String(ModeInfo.szModeName));
// Одержуємо інші
aTTSEnum.Next(1, ModeInfo, @NumFound);
end;
if Speak_Combobox.Items.Count=-1 then
begin
Speak_Combobox.Enabled:=false;
end;
except
Speak_Combobox.Enabled:=false;
end;

// Читання настроювань із ini файлу й розшифровка пароля IRC_NET
Ini:=TiniFile.Create(ExtractFilePath(ParamStr(0))+ 'Config.ini');
// Положення форми на робочому столі
Form1.Left:=Ini.ReadInteger('Position', 'Left', Form1.Left);
Form1.Top:=Ini.ReadInteger('Position', 'Top', Form1.Top);
// Speak
Speak_Combobox.ItemIndex:=Ini.ReadInteger('Speak', 'Speak_Combobox', Speak_Combobox.ItemIndex);
//if (Speak_Combobox.ItemIndex=-1) and (Speak_Combobox.Items.Count>0) then
Speak_Combobox.ItemIndex:=0;
//Speak_ComboboxChange(nil);
// IRC_NET
IRC_net_Nik_Edit.Text:=Ini.ReadString('IRC_NET', 'Nick', IRC_net_Nik_Edit.Text);
IRC_net_Server_Address_Combobox.Text:=Ini.ReadString('IRC_NET', 'Server', IRC_net_Server_Address_Combobox.Text);
IRC_net_Password_Edit.Text:=Ini.ReadString('IRC_NET', 'Password', IRC_net_Password_Edit.Text);
// Розшифровка пароля IRC_NET
s:=IRC_net_Password_Edit.Text;
for i := 1 to Length(s) do s[i] := Char(23 xor Ord(c[i]));
IRC_net_Password_Edit.Text:=s;
Ini.Free;
end;

// Збереження настроювань в ini файлі й шифрування пароля IRC_NET
procedure TForm1.FormDestroy(Sender: TObject);
var
Ini : Tinifile; //необхідно створити об'єкт, щоб потім з ним працювати
// Шифрування пароля IRC_NET
s : String[255];
c : array[0..255] of Byte absolute s;
i : Integer;
begin
// Шифрування пароля IRC_NET
s:=IRC_net_Password_Edit.Text;
for i := 1 to Ord(s[0]) do c[i] := 23 xor c[i];
IRC_net_Password_Edit.Text:=s;

// Запис настроювань в ini файл і розшифровка пароля IRC_NET
Ini:=TiniFile.Create(ExtractFilePath(ParamStr(0))+ 'Config.ini');
// Положення форми на робочому столі
Ini.WriteInteger('Position', 'Left', Form1.Left);

```



```
        aTTSEnum.Select(fpModeInfo^.gModeID, fITTSCentral, IUnknown(fIAMM));  
    except  
    end;  
end;  
  
end.
```

КБПЗ_2023