

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

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

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи обміну файлами
з використанням файлообмінного протоколу BitTorrent”

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

Керівник проекту
доктор філософії (PhD)
_____ Усік П.С.
« ____ » _____ 2023 р.
Рецензент _____

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

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 6 » вересня 2023 року

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

Сіненку Денису Костянтиновичу

(прізвище, ім'я, по батькові)

1. Тема роботи

Дослідження та програмна реалізація системи обміну файлами з використанням файлообмінного протоколу BitTorrent

2. Керівник роботи

Усік Павло Сергійович, доктор філософії (PhD)

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

затверджені наказом вищого навчального закладу № 34-13 від 04.08.2023 року

3. Строк подання студентом роботи до захисту 10.12.2023 р.

4. Мета та завдання випускної кваліфікаційної роботи: *Метою розробки є дослідження та програмна реалізація системи обміну файлами з використанням файлообмінного протоколу BitTorrent*

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

Сіненко Д.К. Дослідження та програмна реалізація системи обміну файлами з використанням файлообмінного протоколу BitTorrent. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2023.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи обміну файлами з використанням файлообмінного протоколу BitTorrent.

Метою розробки є дослідження та програмна реалізація системи обміну файлами з використанням файлообмінного протоколу BitTorrent.

Об'єктом дослідження є процес обміну файлами з використанням файлообмінного протоколу BitTorrent.

Предметом дослідження є методи обміну файлами з використанням файлообмінного протоколу BitTorrent.

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

Результат роботи – програмна реалізація системи обміну файлами з використанням файлообмінного протоколу BitTorrent.

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 Python.

Ключові слова: комп'ютерна інженерія, файлообмінний протокол, BitTorrent

ABSTRACT

Sinenko D.K. Research and software implementation of a file sharing system using the BitTorrent file sharing protocol. 123 Computer engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2023.

In this graduation thesis for the second (master's) level of higher education, software is developed, which is intended for a file exchange system using the BitTorrent file exchange protocol.

The purpose of the development is research and software implementation of a file sharing system using the BitTorrent file sharing protocol.

The object of the research is the file sharing process using the BitTorrent file sharing protocol.

The subject of the study is file sharing methods using the BitTorrent file sharing protocol.

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

The result of the work is a software implementation of a file exchange system using the BitTorrent file exchange protocol.

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 Python environment.

Keywords: computer engineering, file exchange protocol, BitTorrent

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	7
1.1 Призначення системи.....	7
1.2 Область застосування.....	7
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	10
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	10
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	19
2.3 Розгорнута постановка завдання	22
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	23
3.1 Опис функціонування системи	23
3.2 Розробка структурної схеми.....	27
3.3 Розробка функціональної схеми	32
3.4 Розробка діаграми процесів.....	36
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	38
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	38
4.2 Захист розробленого програмного забезпечення.....	48
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	50
6 НАУКОВА НОВИЗНА	52

						ВКРМ-123.23.0019.00.00.ПЗ		
Вим.	Арк.	№ докум.	Підп.	Дата	Дослідження та програмна реалізація системи обміну файлами з використанням файлообмінного протоколу BitTorrent	Літ.	Аркуш	Аркушів
Розроб.	Сіненко Д.К.					М	1	92
Перев.	Усік П.С.							
Н.контр.	Коваленко А.С.					ЦНТУ КІ-22М-1		
Затв.	Смірнов О.А.							

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	53
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	53
7.2 Розрахунок трудомісткості розробки програмної продукції.....	55
7.3 Визначення чисельності виконавців і планового фонду зарплати.....	57
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника.....	61
7.5 Визначення собівартості розробки та ціни програмної продукції.....	66
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	69
7.7 Визначення експлуатаційних витрат.....	70
7.8 Визначення економічної ефективності програмної продукції.....	71
7.9 Висновок.....	73
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	74
8.1 Вступ.....	74
8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером.....	75
8.3 Розробка заходів з умов поліпшення охорони праці.....	77
8.4 Пожежна безпека.....	78
8.5 Розрахунок занулення.....	81
9 ОСНОВНІ ВИСНОВКИ.....	84
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	86

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

- DHT – розподілена геш-таблиця
- HTTP – мережний протокол
- NAT – перетворення мережних адресів
- RSS – стрічка новин
- SHA-1 – геш-функція
- TCP – транспортний протокол
- UDP – транспортний протокол

КБГПЗ-2023

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

Актуальність теми. У сучасному світі дуже широко розповсюджені файлообмінні мережі. Однією з можливих технологій реалізації таких мереж є використання протоколу BitTorrent. BitTorrent – пірінговий (P2P) мережний протокол для кооперативного обміну файлами через Інтернет.

Файли передаються частинами, кожний torrent-клієнт, одержуючи (завантажуючи) ці частини, у той же час віддає (закачує) їхнім іншим клієнтам, що знижує навантаження й залежність від кожного клієнта-джерела й забезпечує

Існує безліч інших програм-клієнтів для обміну файлами за протоколом BitTorrent. Роздача може містити як один файл, так і трохи, наприклад, уміст директорії. Для кожної роздачі створюється файл метаданих з розширенням.torrent, що містить наступну інформацію:

- URL трекера;
- загальну інформацію про файли (ім'я, довжину та ін.) у даній роздачі;
- контрольні суми (точніше, геш-суми SHA1) сегментів файлів, що роздаються;

- Passkey користувача, якщо він зареєстрований на даному трекері. Довжина ключа встановлюється трекером;

- геш-суми файлів цілком (необов'язково);
- альтернативні джерела, що працюють не за протоколом BitTorrent.

Найпоширеніша підтримка так званих web-сидів (протокол HTTP), неприпустимими також є ftp, ed2k, magnet URI (необов'язково).

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи обміну файлами з використанням файлообмінного протоколу BitTorrent.

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

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

– Огляд існуючих систем обміну файлами з використанням файлообмінного протоколу BitTorrent.

– Дослідження системи обміну файлами з використанням файлообмінного протоколу BitTorrent.

– Програмна реалізація системи обміну файлами з використанням файлообмінного протоколу BitTorrent.

Об'єктом дослідження є процес обміну файлами з використанням файлообмінного протоколу BitTorrent.

Предметом дослідження є методи обміну файлами з використанням файлообмінного протоколу BitTorrent.

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 обміну файлами з використанням файлообмінного протоколу BitTorrent.

– Розроблено вітчизняний продукт обміну файлами з використанням файлообмінного протоколу BitTorrent, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі обміну файлами з використанням файлообмінного протоколу BitTorrent.

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 обміну файлами з використанням файлообмінного протоколу BitTorrent, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

КБГПЗ - 2023

					VKPM-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Торренти – це спосіб розповсюдження файлів через Інтернет. Вони працюють через протокол BitTorrent, щоб полегшити так званий одноранговий (P2P) обмін файлами.

Обмін файлами за допомогою торрент-файлів має низку переваг перед традиційним обміном файлами. Для надсилання файлів багатьом людям одночасно не потрібне дороге серверне обладнання, а мережі з низькою пропускною здатністю (повільні) можуть так само легко завантажувати великі набори даних.

Найпоширенішим способом використання торрентів є спеціальний файл із розширенням .TORRENT. У файлі є вказівки щодо того, як ділитися певними даними з іншими людьми.

1.2 Область застосування

Перш ніж ми дізнаємося більше про роботу торрентів, дуже важливо зрозуміти, що вони також становлять більший ризик порівняно з іншими формами обміну файлами.

Торренти за своєю суттю не є небезпечними для використання чи створення, але важливо пам'ятати, що якщо ви не можете довіряти джерелу, дуже легко випадково завантажити файли, які не були завантажені з відповідним юридичним дозволом, або навіть завантажити файли, заражені шкідливим програмним забезпеченням.

Якщо ви зацікавлені у використанні торрентів для обміну власними файлами або для завантаження великих файлів від інших людей, будьте в безпеці

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

2. Імпортуйте файл TORRENT, щоб почати процес завантаження.

3. Збережіть файл на комп'ютері.

У цьому сценарії дані, які ви завантажувате через торрент, можуть існувати на сотнях серверів одночасно, але ці сервери майже завжди є стандартним персональним комп'ютером у домі, як і ваш. Розширене обладнання не потрібне, і кожен може стати учасником цього типу обміну файлами. Насправді кожен, хто завантажує хоча б частину файлу, тепер може працювати як власний торрент-сервер.

Якщо ви зацікавлені в обміні файлами зі свого комп'ютера через Інтернет, традиційним способом зробити це є завантаження даних у центральне розташування (достатньо будь-якого методу обміну файлами), після чого одержувачі зможуть завантажити їх. З торрентами обмін подібний до збереження, як описано вище: замість того, щоб завантажувати торрент, створений кимось іншим і який має вказівки для збереження їхніх файлів, ви надаєте доступ до створеного вами торрента, щоб одержувачі мали необхідні вказівки для завантаження від вас.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи обміну файлами з використанням файлообмінного протоколу BitTorrent, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

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

Проведемо огляд існуючих торрент-систем. Класифікацію проведемо за часом їхньої роботи.

Торрент-системи зі стажем більше року:

1. Uploaded.net – він же Ul.to, Равлик. Як і раніше залишається еталоном стабільності серед торрент-систем. Підтримує високі тарифи й, незважаючи на затримки, що трапляються зрідка, регулярні виплати.

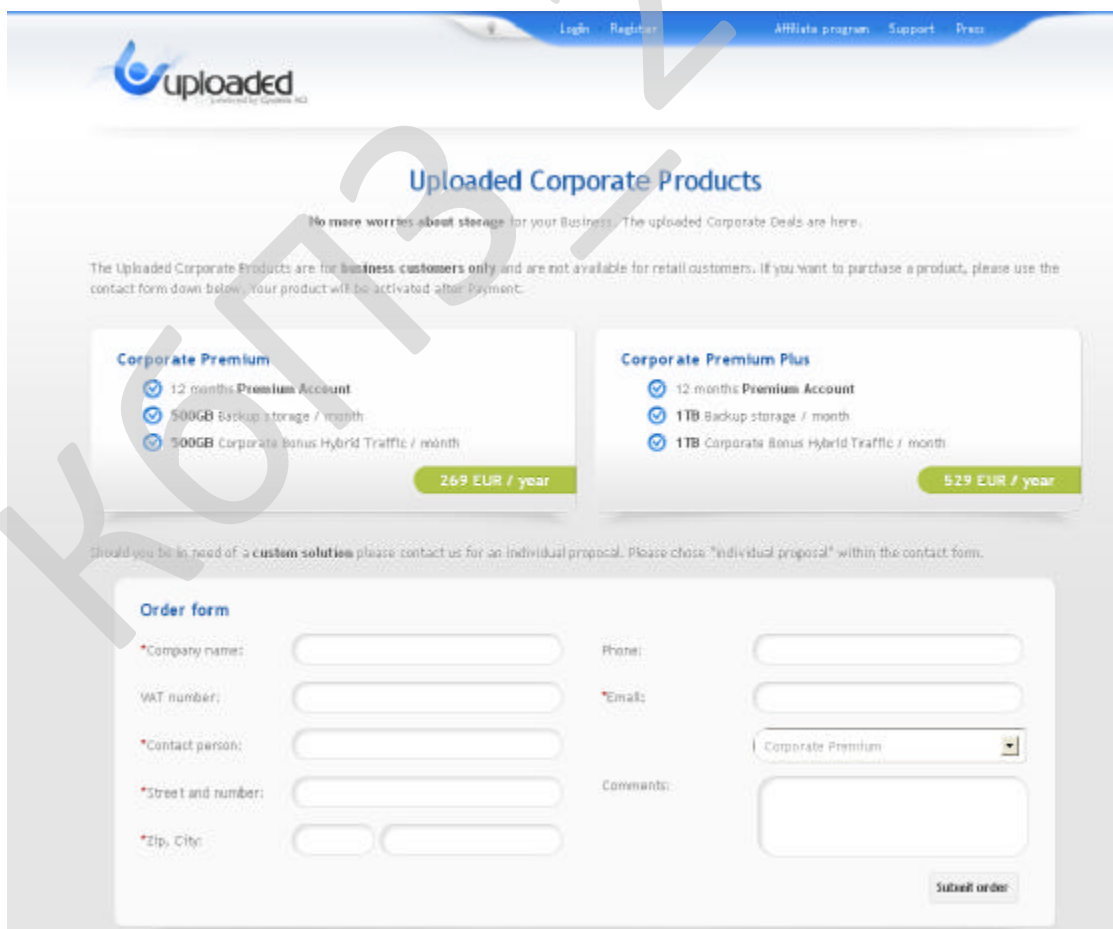


Рисунок 2.1 – Інтерфейс користувача Uploaded.net

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

Одна з далеко не всіх торрент-систем, що настільки довго підтримує оплату завантажень. Маючи власний мерчант, він також не залежить від основної проблеми більшості торрент-систем – складних ситуацій із платіжними системами. Якби не деякі хитрості партнерської програми, що знижують дохід вебмастерів, був би безсумнівним лідером.

2. Depositfiles.com – визнаний ветеран бізнесу торрент-системи, що усе ще має партнерську програму й дозволений практично на кожному Інтернет-ресурсі, що працює з торрент-системами.



Рисунок 2.2 – Інтерфейс користувача Depositfiles.com

На жаль, через низькі тарифи й відсутність PPS (оплати за продажі преміум-аккаунтів), а головне – через тверду політику у відношенні вебмастерів, що порушують DMCA – мало кому підходить для роботи.

3. Hotfile.com – теж ветеран бізнесу, але із ще більш твердою політикою відносно порушень авторських прав. Однак тут скарги ретельно перевіряються, а оплата йде за продажі. Платить приблизно раз на місяць. А тому, підійде тільки тим, хто працює з дійсно не об'ємним контентом.

4. Rapidgator.net – ще один явний лідер, дозволений на переважній більшості форумів і сайтів. У порівнянні з перерахованими вище прийшов на ринок досить пізно, але швидко завоював популярність. Незважаючи на численні обвинувачення в не повному обліку продажів і завантажень, залишається одним із самих прибутковими й стабільних, із щотижневими виплатами, гарними платіжними системами й т.д. Багатьом вебмастерам не подобаються його часті нововведення й технічні нюанси. Однак у сполученні з тим, що тиск правовласників на нього ослабнуло (можливо, у зв'язку з такою, що стала більш жорсткою, політикою дотримання DMCA), він залишається одним із самих затребуваних для роботи.

5. Filefactory.com – зненацька для багатьох, ця найстарша торрент-система вирішила знову заявити про себе. Змінивши свою партнерську програму (помітно піднявши тарифи) і розрахувавшись по боргах, вона знову почав набирати деяку популярність на Заході, переважно у власників своїх ресурсів. З недавнього часу ввела виплати на Webmoney і платить приблизно раз на місяць. Наскільки вона перспективна і чи зможе працювати стабільно протягом довгого часу – поки судити зарано. Але зміни в кращу сторону не можуть не радувати.

6. Extabit.com – а от ця торрент-система переживає не кращі часи. Слабкі платіжні системи, затримки виплат, закрита для новачків партнерська програма – все це говорить не на її користь. Проте, вона усе ще працює й комусь навіть приносить дохід.

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

7. Share-online.biz – досить популярна у Європі, але мало підходяща для вебмастерів зі торрент-система з низькими тарифами.

8. Asfile.com – торрент-система – зразковий ровесник Рапідгатора. Його шлях у бізнесі був досить тернистий, і репутація не завжди була відмінною. Проте, зараз це одна із самих стабільних торрент-систем, із щотижневими виплатами, гарними платіжними системами й великою кількістю ресурсів, де він дозволений. З мінусів хіба що не найвищий тариф.

Далі йдуть трохи більш молоді торрент-системи, зі стажем роботи ледве менш року:

9. Luckyshare.net – у цей момент торрент-система переживає чергову кризу із платіжними системами й затримкою виплат. Але працює все так само стабільно, дозволена на всіх основних площадках для заробітку в західному сегменті мережі, і перспективи у неї усе ще не занадто погані. Потрібно віднестись з обережністю до питання поточної роботи на ній, але списувати з рахунків поки рано й, можливо, варто придивитися на майбутнє.

10. Ryushare.com – так зване «в'єтнамське чудо» дотепер працює. Не будучи явним лідером, він, проте, стабільно платить без сильних затримок і багато де дозволений.

11. Lumfile.com – друга в'єтнамська торрент-система, що з'явилася трохи пізніше попередньої. У зв'язку із цим постійно піддається тиску, втрачав базу даних і платіжні системи. Навіть більше тверда політика у відношенні DMCA в останні місяці не змінила відносини ворогів цієї торрент-системи. Втратившись можливості вести підтримку вебмастерів на форумі WJ, він, однак, як і раніше працює й приносить прибуток, як і раніше дозволений на дуже великій кількості популярних ресурсів інтернету. Але перспективи його, навіть найближчі, стали досить мрячні й неоднозначні.

12. Filestay.com – ще одна азіатська торрент-система зі стабільними виплатами й непоганими платіжними системами. Відрізняється тим, що не намагається пробиватися в лідери й стать дозволеним на топових ресурсах. Йому

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

цілком вистачає тихої спокійної роботи на індивідуальних блогах і малих форумах/сайтах. Підійде тим, хто не женеться за надприбутками, а віддає перевагу такому ж розмірному й стабільному стилю роботи.

13. Upstore.net – торрент-система з теж досить неоднозначною історією. Але в даний момент платить, має непогані платіжні системи, хоча й перебуває далеко від лідируючих позицій. Підійде для власників своїх сайтів з не занадто об'ємним контентом.

14. Livefile.org – у недавньому минулому торрент-система з напівприватною партнерською програмою, тепер остаточно вийшов у паблік. Середні тарифи, середні платіжні системи, середня популярність. На даний момент підійде власникам своїх ресурсів з не занадто об'ємним контентом.

Розглянемо торрент-системи:

1. Vorncash.org – мабуть, на сьогоднішній момент лідер. Нехай не по популярності, то по прибутковості вже напевно. Він і його дзеркала із Групторрент-систем заслужено користуються попитом у вебмастерів, що працюють у сегменті країн.

2. Letitbit.net і його суміжні торрент-системи. Якщо судити по популярності й поширеності посилань на нього, то це як і раніше безперечний лідер. Але по прибутковості він скотився вже досить низько й тому не дуже любий вебмастерами. Як, втім, і користувачами, які практично не можуть скачати з його безкоштовно через достаток реклами й усіляких інсталлерів. Його проект Shareflare – як спроба зайняти своє місце на Заході, теж не є особливо популярним по тим ж причинам. Єдиний плюс цієї торрент-системи, крім поширеності – це гарні платіжні системи й стабільні виплати.

3. Vorncash.com – не найкращий варіант, але працювати можна. Виплати йдуть, багато де дозволений.

4. Gigapeta.com і Tenfiles.com – ще один посередня, але стабільна торрент-система.

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

5. Xlget.com – торрент-система-платник для своїх ресурсів і невеликого обсягу файлів із затребуваним контентом. Щільно зайняв свою вузьку нішу й, фактично не маючи конкурентів, працює не квапливо й стабільно.

6. Exclusivefaile.com – маловідома, але по своєму непогана торрент-система з нормальними тарифами й підходящими платіжними системами.

7. Sinhro.net – довгий час мав проблеми з виплатами, але нові хазяї, здається, цю проблему вирішили. Платить приблизно раз у тиждень зараз, непогані тарифи й гарні платіжні системи. На жаль, маловідомий і лідерам не конкурент.

8. Turbobit.net – за традицією наприкінці списку торрент-систем, але аж ніяк не тому, що гірше інших. По популярності він займає впевнене друге місце після Летитбита, а по прибутковості – після Vorncash.org. Справа ж у тому, що ця торрент-система непогано підходить і для Заходу, хоча мало, де там дозволена. І нехай у неї не найкращі для Заходу платіжні системи, покупки по них є й там. Ну а головні переваги – стабільна робота й виплати – що в нинішній час зовсім непогано. Недавно ввів новий тариф Мікс (PPD+PPS), чим став схожий на Равлика.

Торрент-системи зі стажем приблизно півроку. Тобто вже перевірені часом і труднощами, з регулярними виплатами й, у цей момент стабільною роботою:

1. Keep2share.cc – безперечний лідер у цій категорії огляду, більше того, що досить успішно просувається в лідери бізнесу торрент-систем в цілому. Гарні платіжні системи, стабільні виплати, грамотна й чуйна техпідтримка, дозволяння вже на багатьох популярних ресурсах західного сегмента – все це робить його досить привабливим для роботи й популярним у вебмастерів.

2. Secureupload.eu – торрент-система, дозволена на парі топових ресурсів, з гарними платіжними системами й стабільними виплатами. На жаль, популярності поки не знайшов, так як досить сильно обмежує вебмастерів (висока мінімалка, зберігання файлів не більше 30 днів навіть у преміумів) і ще сильніше

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

3. Filegag.com – торрент-система-аутсайдер, що знову почала випробовувати проблеми з виплатами, але усе ще тримається «на плаву».

4. Depfile.com – торрент-система із сильно підпорченою репутацією, клонині закритого I-filez.com. У даний момент платить і навіть розраховується по старих боргах, але перспективи вкрай сумнівні.

5. Cloudzer.net – по всіх ознаках, другий проект Равлика (Uploaded.net), але суцього для європейських вебмастерів, так як платить зараз тільки на європейські банківські аккаунти й ваучерами Амазона.

6. Netload.in – одна із самих старих торрент-систем, нині платить тільки на європейські банківські аккаунти й із затримкою.

Розглянемо ще більш молоді, але перспективні, і в цей момент такі, що регулярно платять, торрент-системи:

1. Datafile.com – досить перспективна торрент-система з, мабуть, кращим на сьогоднішній день функціоналом. Щотижневі стабільні виплати й чуйна техпідтримка роблять її відмінною торрент-системою для власників своїх ресурсів і постерів по малих форумах.

2. Lafiles.com – проста і зручна торрент-система на всім відомому скрипті з гарними платіжними системами й регулярними виплатами. Аналогічно попередньому – гарний вибір для власників сайтів і блогів.

3. JunoCloud.me – одна із кращих торрент-систем для тих, хто працює на завантаженнях у західному сегменті мережі. Досить високий тариф і щоденні, на сучасний момент, виплати (у тому числі на Webmoney) роблять його прекрасним джерелом заробітку для тих, хто з якихось причин не планує працювати на тарифах PPS.

4. Sanshare.com – у цей момент, як і попередній, більше підійде для тих, хто заробляє на завантаженнях на західних ресурсах. Також щоденні поки виплати на Webmoney і мінімалка всього 5\$. Але тут перспективи є й для тих, хто звик працювати за продажі. Тариф PPS є вже зараз і, у перспективі, якщо у

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

власника вийде підключити гарні платіжні системи, торрент-система стане досить вигідною для всіх.

5. Zefile.com – другий проект торрент-систем Lumfile.com для власників своїх ресурсів. Має всі плюси й мінуси основного проекту. Перспективи по відомих причинах також мрячні, але в цей момент платить і тарифи навіть вище, ніж на Lumfile.com.

6.Ex-load.com – досить незвичайна молода торрент-система, схожа на Турбобит, але яка пішла ще далі у своїй спробі стати універсальною для країн і Заходу. Фактично це дві різних торрент-системи в одній, залежно від того, під яким ір Ви заходите на його платіжну сторінку або сторінку завантаження. Наскільки він буде стабільн і вигідним – покаже час, але перспективи гарні.

7. Brutalsha.re – французька торрент-система, що представляє собою майже точну копію багатьом пам'ятного 1hostclick.com. Тобто сполучена партнерська система PPD+PPS, орієнтована на Захід, приблизно з тими ж тарифами. Виплати на PayPal.

8. Oceanus.ch – торрент-система з гарними тарифами, що включають високий відсоток за продажі із сайту. Немає виплат на Webmoney, але є на Liberty Reserve, щотижневі.

9. Ufoх.com – торрент-система з непоганими тарифами PPS і PPD, але, на жаль, що видаляє заробіток з файлів, що порушують DMCA.

10. Hugefiles.net – уже перевірена часом торрент-система для спамерів, постящих погано продаваний, але нормально завантажуваний контент. Український тариф компенсується обліком практично всіх завантажень.

11. Filezy.net – приблизно те ж саме, що попередній, тільки більш новий й немає виплат на WebMoney. Для аматорів постить на азіатські й латиноамериканські форуми підійде. Для них, до речі, з'явився й російськомовна торрент-система з виплатами на WebMoney й ледве більше високими тарифами – Thefile.me

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

Існують ще три торрент-системи, які я утрудняюся віднести до якоїсь групи:

1. Filesmonster.com – досить стара торрент-система, але дуже специфічна. Щільно співробітничав із правовласниками, у результаті завантаження контенту закрито й можна лише роздавати файли з каталогу.

2. Fileparadox.in – досить непогана торрент-система з гарними тарифами (у тому числі PPD) і платіжними системами. Однак недавно сайт піддався атаці хакерів, був зламаний і тепер перспективи невизначені, хоча вони й намагаються відновити файли.

3. Fiberstorage.net – турецька або арабська торрент-система із приватною партнерською програмою, що відкривається після співбесіди (на щастя, на англійській).

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Як мова програмування обрана Python. Python – високорівнева мова програмування загального призначення з акцентом на продуктивність розроблювача й читаність коду. Синтаксис ядра Python мінімалістичний. У той же час стандартна бібліотека включає великий обсяг корисних функцій.

Python підтримує кілька парадигм програмування, у тому числі структурне, об'єктно-орієнтоване, функціональне, імперативне й аспектно-орієнтоване. Основні архітектурні риси – динамічна типізація, автоматичне керування пам'яттю, повна інтроспекція, механізм обробки виключень, підтримка багатопоточні обчислень і зручні високорівневі структури даних. Код у Пітоні організовується у функції й класи, які можуть поєднуватися в модулі (які у свою чергу можуть бути об'єднані в пакети).

Еталонною реалізацією Python є інтерпретатор CPython, що підтримує більшість активно використовуваних платформ. Він поширюється вільно під

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

дуже ліберальною ліцензією, що дозволяє використовувати його без обмежень у будь-яких застосунках, включаючи пропрієтарні. Є реалізації інтерпретаторів для JVM (з можливістю компіляції), MSIL (з можливістю компіляції), LLVM і інших. Проект PyPy пропонує реалізацію Пітона на самому Пітоні, що зменшує витрати на зміни мови й постановку експериментів над новими можливостями.

Python – активно, що розвивається мова, програмування, нові версії (з додаванням/зміною мовних властивостей) виходять приблизно раз у два з половиною року. Внаслідок цього й деяких інших причин на Python відсутні ANSI, ISO або інші офіційні стандарти, їхню роль виконує CPython.

Python портований і працює майже на всіх відомих платформах – від КПК до мейнфреймів. Існують порти під Microsoft Windows, практично всі варіанти UNIX (включаючи FreeBSD і Linux), Plan 9, Mac OS і Mac OS X, iPhone OS 2.0 і вище, Palm OS, OS/2, Amiga, AS/400 і навіть OS/390, Symbian і Android.

У міру старіння платформи її підтримка в основній гілці мови припиняється. Наприклад, із серії 2.6 припинена підтримка Windows 95, Windows 98 і Windows ME. Однак на цих платформах можна використовувати попередні версії Python – на даний момент співтовариство активно підтримує версії Python починаючи від 2.3 (для них виходять виправлення).

При цьому, на відміну від багатьох портуємих систем, для всіх основних платформ Python має підтримку характерних для даної платформи технологій (наприклад, Microsoft COM/DCOM). Більше того, існує спеціальна версія Пітона для віртуальної машини Java – Jython, що дозволяє інтерпретаторові виконуватися на будь-якій системі, що підтримує Java, при цьому класи Java можуть безпосередньо використовуватися з Пітона й навіть бути написаними на Пітоні. Також кілька проектів забезпечують інтеграцію із платформою Microsoft .NET, основні з яких – IronPython і Python.Net.

Python підтримує динамічну типізацію, тобто тип змінної визначається тільки під час виконання. Тому замість «присвоювання значення змінної» краще говорити про «зв'язування значення з деяким ім'ям». У Пітоні є убудовані типи:

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

бульові, рядки, Unicode-рядки, цілі числа довільної точності, числа із плаваючою комою, комплексні числа й деякі інші. З колекцій Python підтримує кортежі (*tuples*), списки, словники (асоціативні масиви) і, починаючи з версії 2.4, безлічі. Всі значення в Пітоні є об'єктами, у тому числі функції, методи, модулі, класи.

Додати новий тип можна або написавши клас (*class*), або визначивши новий тип у модулі розширення (наприклад, написаному мовою C). Система класів підтримує спадкування (одиначне й множинне) і метапрограмування. Можливе спадкування від більшості убудованих типів і типів розширень.

Всі об'єкти діляться на посилальні й атомарні. До атомарного ставляться `int`, `long`, `complex` і деякі інші. При присвоюванні атомарних об'єктів копіюється їхнє значення, у той час як для посилальних копіюється тільки покажчик на об'єкт, таким чином, обидві змінні після присвоювання використовують те саме значення. Посилальні об'єкти бувають змінювані й незмінні. Наприклад, рядки й кортежі є незмінними, а списки, словники й багато інших об'єктів – змінюваними. Кортеж у Пітоні є, по суті, незмінним списком. У багатьох випадках кортежі працюють швидше списків, тому якщо ви не плануєте змінювати послідовність, то краще використовувати саме їх.

Мова має чіткий і послідовний синтаксис, продуману модульність й масштабованість, завдяки чому вихідний код написаних на Пітоні програм легко читаємий.

Python – стабільна й розповсюджена мова. Він використовується в багатьох проектах і в різних якостях: як основна мова програмування або для створення розширень і інтеграції застосунків. На Python реалізоване велика кількість проектів, також він активно використовується для створення прототипів майбутніх програм. Python використовується в багатьох великих компаніях.

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 обміну файлами з використанням файлообмінного протоколу BitTorrent.

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

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

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

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

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

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

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

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

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

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Все це може здатися трохи заплутаним, але насправді ідея досить проста. Торренти, як ви читали вище, покладаються на однорангову мережу. Це лише означає, що до торрент-даних, якими б вони не були, можна отримати доступ із кількох серверів одночасно. Кожен, хто завантажує торрент, отримує його частинами з інших серверів.

Я вмикаю торрент і ділюся файлом онлайн. Десятки людей завантажують його, і ти один із них. Ваша торрент-програма вибере, з якого сервера взяти файл, залежно від того, хто наразі ділиться ним і на яких серверах є та частина файлу, яка вам зараз потрібна.

У традиційній системі обміну файлами, яка використовує файловий сервер, надання спільного доступу до програми розміром 200 МБ для 1000 людей швидко вичерпає всю пропускну здатність завантаження, особливо якщо всі запитують файл одночасно. Торренти усувають цю проблему, дозволяючи клієнтам збирати лише трохи даних від мене, трохи від іншого користувача і так далі, доки вони не завантажуть увесь файл.

Коли файл завантажено декількома особами, початковий учасник може припинити його розповсюдження, не вплинувши на інших. Файл залишатиметься доступним для будь-яких інших користувачів цього торрента через децентралізовану P2P-основу BitTorrent.

Як розповсюджуються торренти

Після створення торрента автор може поділитися однією з двох речей: файлом.TORRENT або хешем торрента, який часто називають магнітним посиланням.

Магнітне посилання – це простий спосіб ідентифікації торрента в мережі

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

BitTorrent без необхідності мати справу з файлом TORRENT. Подібно до відбитка пальця, він унікальний для конкретного торрента, тож хоча посилання є лише рядком символів, воно так само добре, як і наявність файлу.

Магнітні посилання та файли TORRENT часто перераховуються в торрент-індексах, які є сайтами, створеними спеціально для обміну торрентами. Ви також можете ділитися інформацією про торрент через електронну пошту, текстове повідомлення тощо.

Оскільки магнітні посилання та файли TORRENT є лише інструкціями для клієнта BitTorrent, щоб зрозуміти, як отримати дані, обмінюватися ними швидко та легко.

Торрент-файл не дуже корисний, якщо він не використовується з клієнтською програмою. Ось приклад торрента, відкритого в текстовому редакторі – ви бачите, наскільки безглуздим є перегляд торрента таким чином.

Загальні умови торрент

Ось кілька корисних термінів, які варто знати під час роботи з торрентами:

– **Сід:** Завантажити торрент означає поділитися ним. Кількість початкових файлів торрента – це кількість людей, які поділилися повним файлом. Нульова кількість означає, що ніхто не може завантажити весь файл.

– **Одноранговий:** одноранговий – це той, хто завантажує файл із розсівача, але ще не має повного файлу.

– **П'явка:** п'явки завантажують більше, ніж завантажують. Натомість п'явка може взагалі нічого не завантажувати після завантаження повного файлу.

– **Swarm:** група людей, які завантажують і діляться одним торрентом.

– **Трекер:** сервер, який відстежує всіх підключених користувачів і допомагає їм знайти один одного.

– **Клієнт:** програма або веб-служба, яка використовується торрент-файлом або магнітним посиланням, щоб зрозуміти, як завантажувати або завантажувати файли.

Як завантажувати торренти без відома провайдера?

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

Використання торрентів є легальним і ефективним способом обміну великими файлами. Хоча інтернет-провайдери не заборонятимуть вам використовувати торренти, вони можуть інколи гальмувати трафік BitTorrent, що сповільнить швидкість завантаження. Якщо ви не хочете, щоб ваш провайдер знав, що ви завантажуєте торренти, вам потрібно буде використовувати віртуальну приватну мережу (VPN), яка захищає вашу конфіденційність.

Як завантажувати торренти за допомогою VPN?

Щоб безпечно завантажувати торренти за допомогою VPN, знайдіть VPN для підтримки P2P, політику «нульового журналювання» (дані сеансу не відстежуються та не зберігаються), «перемикач блокування», який негайно розриває ваше інтернет-з'єднання, якщо VPN-з'єднання втрачено, і високі швидкості. Вибравши постачальника послуг VPN, завантажте та встановіть програмне забезпечення, подбавши про те, щоб використовувати найбезпечніші налаштування. Потім виберіть торрент-сервер із безпечним легальним вмістом, підключіться до VPN і встановіть безпечне з'єднання.

Як транслювати торренти?

Коли ви транслюєте торрент, наприклад, файл фільму, ви зможете дивитися фільм, не чекаючи, поки завантажиться весь файл. Для цього вам знадобиться спеціальний сайт або інструмент для потокової передачі торрент-файлів. Деякі приклади включають WebTorrent Desktop, Webtor.io та Seedr. Однак перед тим, як транслювати будь-який торрент, переконайтеся, що вміст безкоштовний і доступний законно, наприклад, фільм, який є загальнодоступним.

Як зробити торренти швидшими?

Є кілька речей, які ви можете зробити, щоб пришвидшити завантаження торрент-файлів. Спочатку перевірте, скільки є «сідерів» для торрент-файлу. Сідери – це люди, які продовжують ділитися торрентом після того, як його завантажили. Чим більше сідерів, тим швидше будуть завантажувати торрент-файли. Ви також можете спробувати уникати Wi-Fi на користь дротового підключення до Інтернету, завантажувати файли один за одним, обходити

						ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			25

брандмауер або перейти на більш швидкісний тарифний план Інтернету.

BitTorrent

BitTorrent – це одноранговий інструмент обміну медіафайлами, але це не так просто, як завантажувати вміст у браузері. Наш посібник покаже вам, як використовувати Torrent і як убезпечити себе під час цього.

Якщо ви проводили якийсь час в Інтернеті з початку 2000-х років, ви, мабуть, чули про BitTorrent, популярний інструмент для обміну файлами, який використовується для фільмів, програм та інших великих блоків даних. Але BitTorrent є дещо складнішим, ніж інструменти для обміну файлами, які з'явилися раніше, тому перед його використанням варто ознайомитися з його тонкощами. Наш посібник допоможе вам почати роботу.

Що таке BitTorrent?

Коли ви завантажуєте файл з Інтернету, ви зазвичай завантажуєте його безпосередньо з сервера. Швидкість завантаження файлу залежить від розташування сервера, швидкості та кількості людей, які намагаються завантажити файл одночасно. Таким чином, хоча у вас може бути з'єднання 200 Мбіт/с від вашого інтернет-провайдера, ви можете завантажувати файл набагато повільніше, ніж це, якщо сервер, який надає файл, працює повільно або забивається запитами.

BitTorrent – це альтернативний спосіб завантаження великих файлів, який не має тих самих недоліків. BitTorrent – це не просто програма – це одноранговий (p2p) протокол, який може використовувати будь-яка програма (хоча існує програма під назвою BitTorrent, що належить однойменній компанії, яка розробляє та підтримує сам протокол).

Замість того, щоб завантажувати файл з одного сервера, за допомогою BitTorrent ви завантажуєте частини цього файлу від інших користувачів по всьому світу, які мають той самий файл на своєму комп'ютері (отже, *одноранговий*). Файл або група файлів, які ви завантажуєте, називається торрентом, обмін цими файлами називається *завантаженням*, а група людей, від

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

яких ви завантажуєте, називається *роєм*. Чим більше людей підключено до певного рою, завантажуючи файл, тим швидше ви зможете завантажити цей файл.

BitTorrent широко відомий як піратський інструмент завдяки своїй ефективності обміну великими файлами фільмів, музичних альбомів, програмного забезпечення та порно. Використання BitTorrent за своєю суттю не є незаконним, і протокол також бачив багато законних застосувань, зокрема обмін програмним забезпеченням з відкритим кодом, таким як Linux, надання оновлень для таких програм, як World of Warcraft, синхронізація файлів між комп'ютерами у стилі Dropbox та обмін медіафайлами, випущений через BitTorrent самими художниками. Microsoft навіть використовує подібні технології для оптимізації оновлень Windows.

Загалом, BitTorrent часто використовується для піратства, оскільки його ефективність, децентралізований характер і популярність створили жваву спільноту, яка ділиться цими файлами. Якщо є щось, що ви хочете завантажити, є хороший шанс, що хтось поділиться цим із BitTorrent, законно чи ні. Якщо ви використовуєте наші інструкції з цією метою, ви робите це на свій власний ризик.

3.2 Розробка структурної схеми

На рисунку 3.1 зображена структурна схема системи. Вона складається з наступних блоків:

- Вікно інформації про файли, які завантажуються та роздаються з використанням файлообмінного протоколу BitTorrent.
- Вікно рядку стану.
- Блок меню налаштувань.
- Панель інструментів.
- Список категорій.
- Вікно докладної інформації.

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27



Рисунок 3.1 – Структурна схема системи

Розглянемо принципи роботи протоколу BitTorrent.

Як завантажити торрент

1. Отримайте клієнт BitTorrent

Завантажувати файли за допомогою BitTorrent дещо складніше, ніж просто натиснути посилання у веб-браузері. Більшість браузерів не мають вбудованої підтримки BitTorrent, тому вам потрібна спеціальна програма, яка називається клієнтом BitTorrent, яка знає, як завантажувати та збирати фрагменти файлу в торрент.

2. Знайдіть торрент для завантаження

Ви можете знайти торренти для завантаження кількома різними способами. Якщо ви завантажуєте щось, що автор зробив доступним на

BitTorrent, ви, ймовірно, знайдете посилання для завантаження торрент-файлу десь на їхньому сайті – наприклад, дистрибутив Linux Ubuntu пропонує торренти на своїй сторінці альтернативних завантажень. Ви також можете відвідати пошукову систему торрентів, як-от влучно названу Pirate Bay, щоб знайти щось.

3. Завантажте торрент-файл

Коли ви знайдете те, що шукаєте, вам потрібно буде завантажити файл .torrent – крихітний фрагмент даних, який спрямовує ваш клієнт BitTorrent на рій, щоб він міг завантажити фактичні файли, які ви шукаєте. Отже, якщо ви завантажуєте відео, ви спочатку завантажите невеликий файл .torrent, а потім двічі клацніть його, щоб відкрити його в qBittorrent, де почнеться завантаження самого відео.

Це може здатися дещо обхідним шляхом – завантаження невеликого файлу, який спрямовує вас на інший файл, – але це спосіб роботи BitTorrent. Багато сайтів і клієнтів BitTorrent також підтримують «магнітні посилання», які дозволяють обійти файл .torrent і направити клієнта на рій простим клацанням миші. Ви знайдете файли .torrent і магнітні посилання на сторінці завантаження потрібного файлу (якщо автор підтримує BitTorrent) або в пошуковій системі BitTorrent, наприклад Pirate Bay.

4. Відкрийте торрент у своєму клієнтському програмному забезпеченні

Просто двічі клацніть торрент і скажіть своїй ОС відкрити його в клієнтському програмному забезпеченні. Звідти вам потрібно буде вказати програмному забезпеченню, куди завантажити ваш файл, і ви також зможете визначити його пріоритет серед усіх інших речей, які ви хочете завантажити одночасно. Однак це індивідуальні кроки клієнта.

Як безпечно та швидко завантажити торрент

Це найнеобхідніші основи, необхідні для початку роботи з BitTorrent, але вони навряд чи забезпечать вам оптимальний досвід. Нижче наведено кілька речей, які допоможуть вам бути в безпеці та забезпечити швидке завантаження.

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

Остерігайтеся зловмисного програмного забезпечення та підроблених торрентів

Як і будь-де в Інтернеті, ви повинні бути обережними щодо того, що ви завантажуєте. Хоча багато, навіть більшість торрент-файлів містять те, що нібито містять, деякі з них обдурять вас, пообіцявши одне, а замість цього пропонуючи зловмисне програмне забезпечення, особливо якщо ви шукаєте торренти в темніших куточках Інтернету.

Багато торрент-сайтів мають коментарі на кожній сторінці, які можуть допомогти вам визначити автентичність певного торрента на основі написаного іншими користувачами. Деякі каталоги розміщують позначки поруч із іменем користувача завантажувача, позначаючи користувачів із хорошим досвідом завантаження безпечних файлів. І, звісно, ви маєте запуснути хороше антивірусне програмне забезпечення, яке автоматично сканує все, що ви завантажуєте.

Використовуйте VPN

Оскільки ви підключаєтеся безпосередньо до інших користувачів, щоб завантажити свій торрент, ви можете бачити IP-адресу кожного користувача в групі, і вони можуть бачити вашу. Ця IP-адреса простежується до вас, що дозволяє кожному, хто приєднається до рою, бачити, що ви завантажуєте. Деякі інтернет-провайдери, особливо ті, що знаходяться за межами США, також можуть зменшувати вашу швидкість, якщо бачать, що ви використовуєте BitTorrent. Щоб запобігти обом цим проблемам, ви повинні використовувати надійну VPN під час завантаження торрент-файлів – це приховає вашу IP-адресу від зграї та зашифрує ваш трафік, щоб ваш провайдер не знав, що ви робите.

Уникайте мертвих торрентів

BitTorrent дозволяє завантажувати файли швидше та ефективніше, ніж пряме з'єднання з сервером, за умови, що є достатньо велика група, яка ділиться файлом. Якщо торрент має лише пару сідерів, ви, ймовірно, завантажуватимете файл дуже повільно, а якщо сідерів немає, ви взагалі не зможете отримати файл. Коли ви завантажуєте торрент, переконайтеся, що ви завантажуєте останній файл

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

із великою кількістю сідерів, щоб забезпечити найкращу швидкість (багато сайтів вказують кількість сідерів на сторінці завантаження торрента).

Переконайтеся, що ви відкриті для зв'язків

У наші дні більшість торрент-клієнтів налаштовано оптимально з коробки – вони створюють для себе правила брандмауера Windows і використовують UPnP або NAT-PMP для автоматичного відкриття порту, щоб розсівачі могли надсилати вам свої частини файлу. Однак якщо програма має багато засівачів і файл не завантажується, вам, можливо, доведеться вручну перенаправити порт на вашому маршрутизаторі або виконати інші дії з усунення несправностей, щоб переконатися, що ви можете підключитися до однорангових пристроїв у групі.

Дотримуйтеся етикету BitTorrent

Хоча це не має вирішального значення для успіху, загалом вважається приємним перераховувати кошти та створювати початкові файли після їх завантаження, даючи іншим можливість завантажувати дані самостійно. Зрештою, якби не було сідерів, BitTorrent не працював би дуже добре, і ми всі застрягли б у завантаженні файлів із швидкістю равлика. Якщо ви не можете завантажувати файл вічно, принаймні заповнюйте його, доки ви не поділитесь стільки, скільки завантажили (також відоме як "співвідношення" 1:1), перш ніж видаляти торрент із вашого клієнта.

Тут є ще більше, у що ми можемо заглибитися, оскільки BitTorrent є надзвичайно потужним інструментом, якщо ви бажаєте копатися в налаштуваннях клієнта. Але для більшості початківців це повинно допомогти вам почати роботу, убезпечити вас і дозволити вам уникнути страшного «повільного сервера завантаження», наскільки це можливо.

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

3.3 Розробка функціональної схеми

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

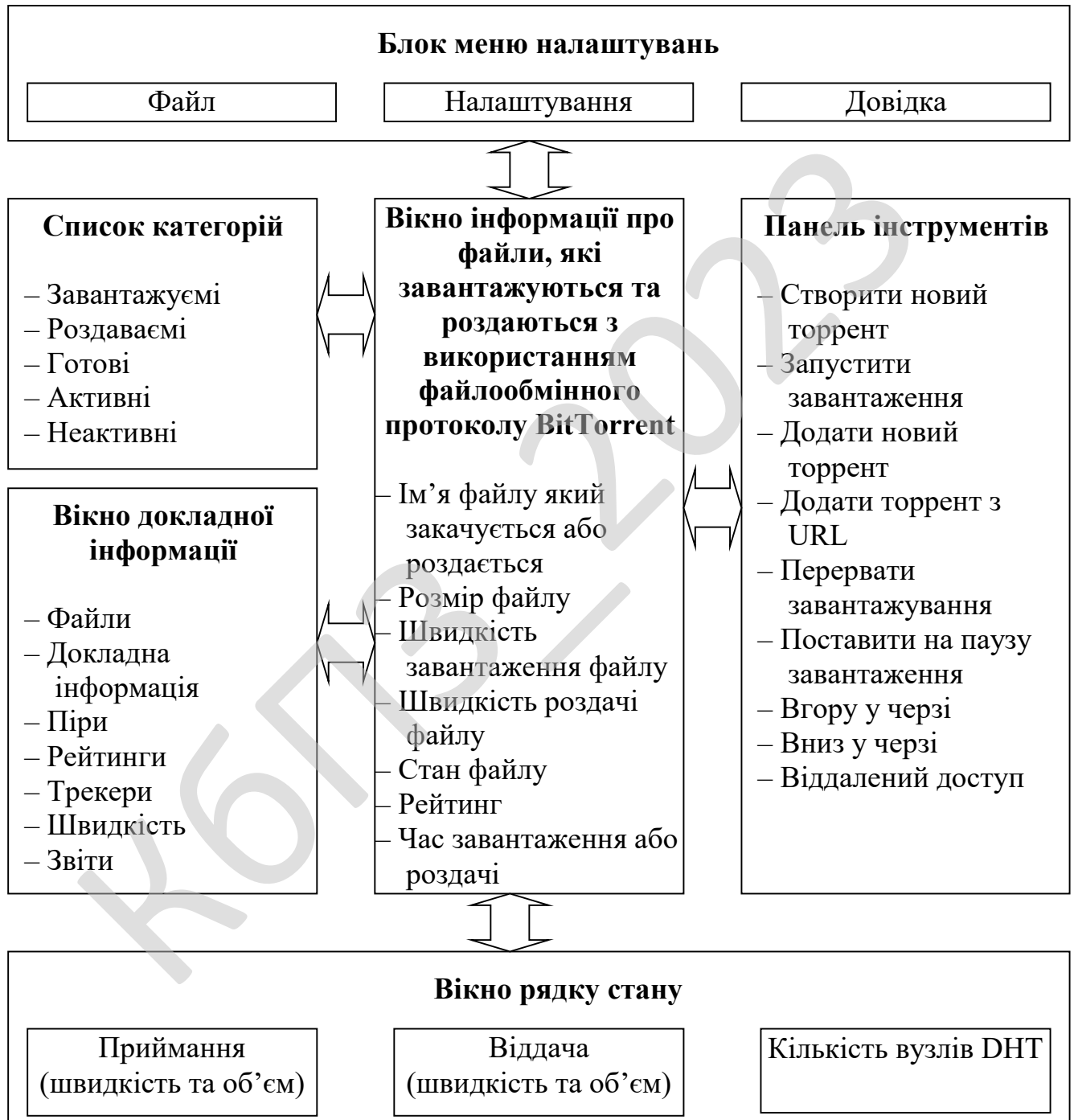


Рисунок 3.2 – Функціональна схема системи

Приведемо функціональне призначення наведених блоків:

1. Вікно інформації про файли, які завантажуються та роздаються з використанням файлообмінного протоколу BitTorrent:

- Ім'я файлу який закачується або роздається.
- Розмір файлу.
- Швидкість завантаження файлу.
- Швидкість роздачі файлу.
- Стан файлу.
- Рейтинг.
- Час завантаження або роздачі.
- № файлу для завантаження або роздачі.

2. Блок меню налаштувань:

– Файл. Він включає в себе наступні елементи: Додати торрент, Додати торрент з URL, Додати RSS– ленту, Створити новий торрент, Вихід.

– Налаштування. Він включає в себе наступні елементи: Конфігурація, RSS-завантажувач, Помічник установки, Відображення різних панелей (з клавішами швидкого доступу), Значки на вкладках, Завершення сеансу Windows.

– Довідка. Він включає в себе наступні елементи: Довідка, Типові питання та відповіді, Перевірка оновлення, Статистика, про програму.

3. Список категорій:

- Завантажуємі.
- Роздаємі.
- Готові.
- Активні.
- Неактивні.

4. Вікно докладної інформації:

- Файли.
- Докладна інформація.
- Піри.
- Рейтинги.
- Трекери.

– Швидкість.

– Звіти.

5. Панель інструментів:

– Створити новий торрент.

– Запустити завантаження.

– Додати новий торрент.

– Додати торрент з URL.

– Перервати завантажування.

– Поставити на паузу завантаження.

– Вгору у черзі.

– Вниз у черзі.

– Віддалений доступ.

6. Вікно рядку стану:

– Приймання (швидкість та об'єм).

– Віддача (швидкість та об'єм).

– Кількість вузлів DHT.

BitTorrent-трекер реалізований на одному з популярних движків (російськомовних і не тільки) ТоррентРієр. В основі організації лежить принцип торрент → топик, тобто торренту зіставлена єдина тема на форумі. Torrent-файл завантажується в перше повідомлення теми автором, після чого відбувається реєстрація цього торренту на трекері. Варто врахувати, що хоча форумна й трекерна частини досить тісно інтегровані, трекерна частина використовує окремі таблиці в БД, тому можливо відокремити трекерну частину (анонсер) і винести її, приміром, на окремий сервер.

TBDev

Розроблявся переважно для каталогу torrentbits.org, згодом був викладений під GNU General Public License. Виконаний як система керування вмістом. Має безліч модифікацій.

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

Опис .torrent-файлу

Файл метаданих є словником в bencode форматі з розширенням .torrent – використовується в р2р мережі BitTorrent і містить інформацію про файли, трекерах і ін.

Містить наступну інформацію:

- URL трекеру.
- Загальну інформацію про файли (ім'я, довжину та ін.) у даній роздачі.
- Контрольні суми (точніше, геш-суми SHA-1) сегментів файлів, що роздаються.
- Passkey користувача, якщо він зареєстрований на даному трекері. Довжина ключа встановлюється трекером.
- (Необов'язково) геш-суми файлів цілком.
- (Необов'язково) Альтернативні джерела, що працюють не за протоколом BitTorrent. Найпоширеніший підтримку так званих web-сідів (протокол HTTP), але припустимими також є ftp, ed2k, magnet URI.

Розмір сегмента регулюється при створенні торренту й, як правило, вибирається розмір, що відповідає ступені двійки. При виборі розміру необхідно дотримувати балансу, пов'язаний з механізмом роботи протоколу.

Розмір сегмента найчастіше лежить у діапазоні від 128 кілобайт до 2-4 мегабайт, хоча на дуже більших роздачах (порядку сотні гігабайт) можуть використовуватися сегменти розміром 32-64 мегабайта.

Якщо роздача складається з декількох файлів, то в процесі гешування вони зчитуються підряд і розглядаються як безперервний потік даних.

Тому найчастіше сегмент, що містить кінець одного файлу, також містить і початок наступного.

Разом з тим для того, щоб переконатися в правильності завантаженого сегмента, необхідно мати його весь цілком.

Саме тому, незважаючи на те, що більшість клієнтів підтримує завантаження не всіх файлів у роздачі, а тільки деяких, майже завжди буде

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

завантажений також і початковий і/або кінцевий шматок файлів, не обраних для завантаження.

Так як геши в .torrent-файлі містять у собі імена й структуру директорій роздачі, то перейменування файлів зі збереженням можливості їх роздавати в загальному випадку неможливо.

Однак, деякі клієнти підтримують зміну структури, наприклад, створення або перейменування директорій і перейменування або переміщення файлів.

Файл метаданих є словником в bencode форматі. Файли метаданих можуть поширюватися через будь-які канали зв'язку: вони (або посилання на них) можуть викладатися на веб-серверах, розміщатися на домашніх сторінках користувачів мережі, розсилатися по електронній пошті, публікуватися в блогах або новиних стрічках RSS.

Також є можливість одержати info частина публічного файлу метаданих прямо від інших учасників роздачі завдяки розширенню протоколу «Extension for Peers to Send Metadata Files».

Це дозволяє обійтися публікацією тільки магнет-посилання. Одержавши яким-небудь образом файл із метаданими, клієнт може починати завантаження.

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

3.4 Розробка діаграми процесів

Діаграма взаємодії процесів системи, розробленої у результаті виконання магістерського проектування, наведена на рисунку 3.3.

Після початку роботи проходить виведення вікна програми, звідки можна провести налаштування системи та відправлення даних чи прийом.

Якщо обирається відправлення даних проходить моніторинг торренту (сіди/піри) та формування пакету даних з відправленням пакету даних та отриманням службової інформації.

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

На рисунку 4.1 наведено блок-схему основної програми. Її робота складається з виконання наступних кроків. Спершу відбувається виведення основного вікна програми.

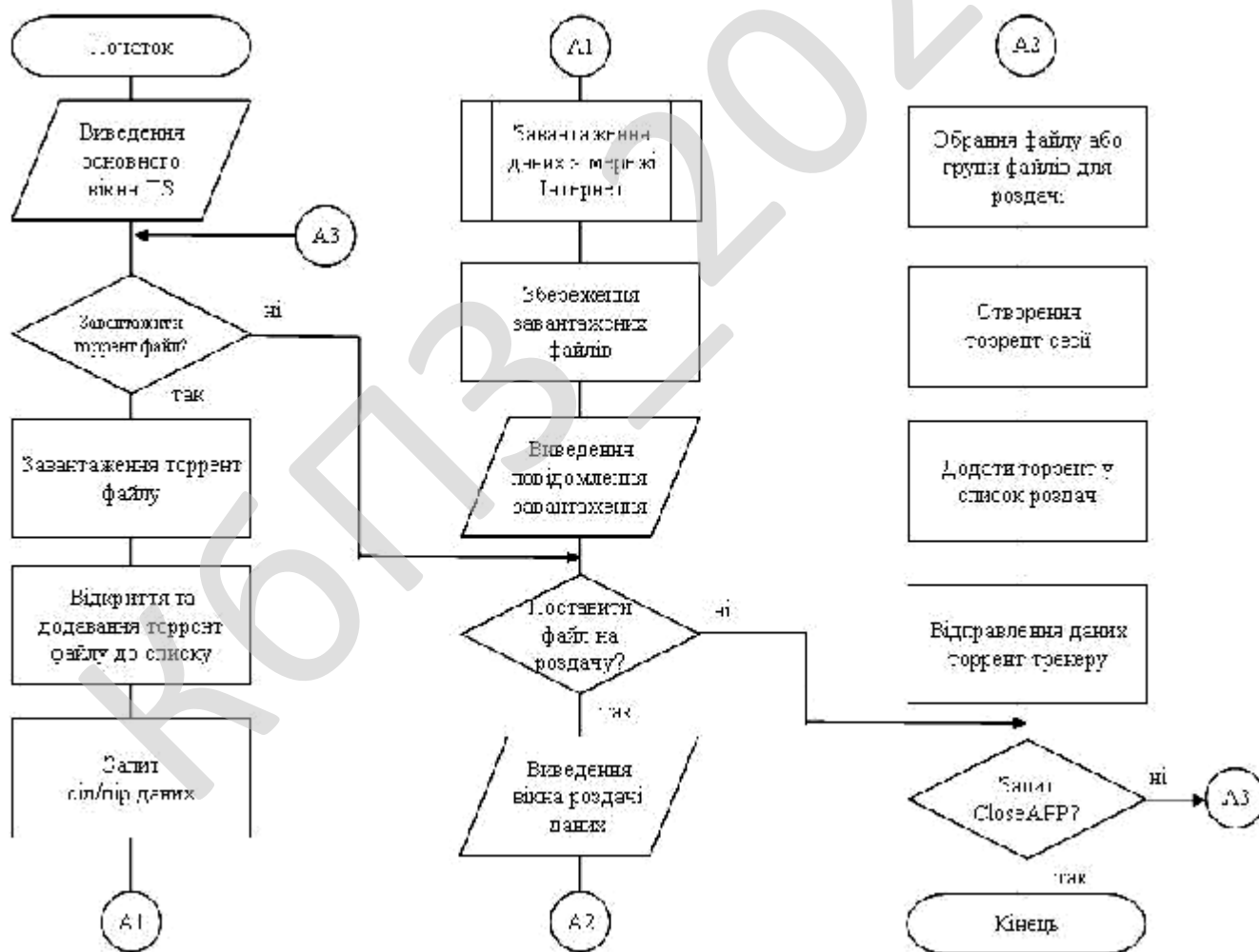


Рисунок 4.1 – Блок-схема основної програми

Після цього проходить запит завантаження торрент файлу для отримання фалів з Інтернет мережі. Якщо такий запит є, проводиться завантаження торрент файлу, відкриття та додавання торрент файлу до списку, запит сід/пір даних та виконання підпрограми завантаження даних з мережі Інтернет. У подальшому після виконання підпрограми проходить збереження завантажених файлів, та виведення повідомлення завантаження даних.

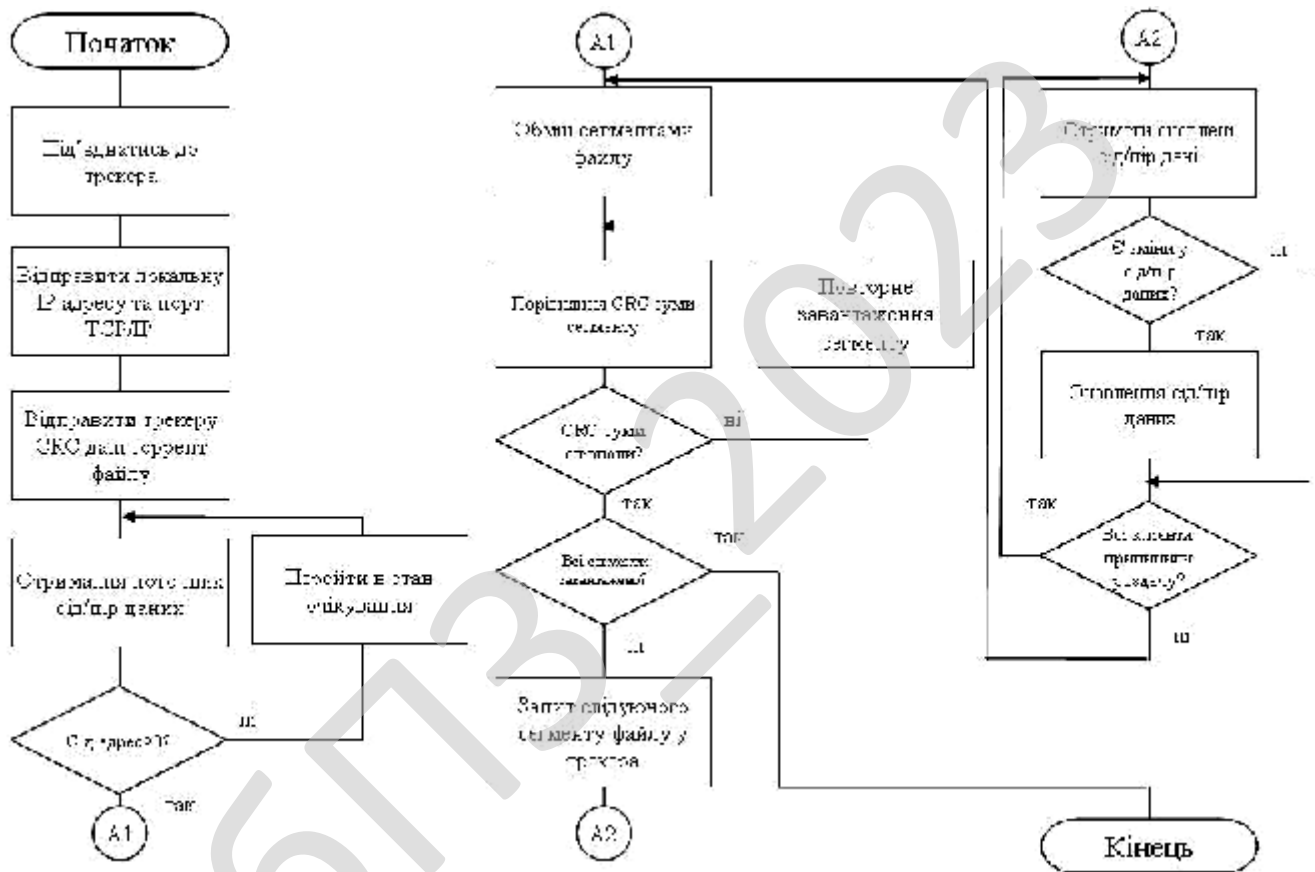


Рисунок 4.2 – Блок-схема підпрограми завантаження файлу

Далі проходить запит – поставити файл на роздачу. Якщо запит позитивний проводиться виведення вікна роздачі даних, обрання файлу або групи файлів для роздачі, створення торрент сесії, додавання торренту у список роздач та відправлення даних торрент трекеру. Якщо поступає сигнал CloseAPP проводиться завершення роботи програми.

На рисунку 4.2 зображено роботу підпрограми завантаження файлу. Після вхідної точки проводиться під'єднання до трекера, відправка локальну IP адресу та порт TCP/IP, відправка трекеру CRC дані торрент файлу.

Після цих ініціалізацій них ді проводиться отримання поточних сид/пір даних та якщо уточнюється що немає жодного Сіда проводиться чекання – перехід в стан очікування. Коли сид з'являється проходить обмін сегментами файлу, порівняння CRC суми сегменту. Якщо CRC суми співпали проходить запит на перевірку – всі сегменти завантажено. Тобто завантажено чи ні файл з Інтернет мережі. Якщо ні проводиться запит слідуючого сегменту файлу у трекера, отримання оновлених сид/пір даних. Ці дії проходять з перевітками та оновленням змін сид/пір даних.

Торрент клієнт

Розробка торрент клієнту – складна справа. Необхідно спочатку досконально зрозуміти як працює протокол обміну, як проходить сегментація та розбиття файлу на сегменти та багато іншого.

Розроблене ПЗ було написано на мові Python з додаванням модулів та функцій C# та бібліотеки Monotorrent.

Monotorrent – одна з найбільш професійних і функціональних серед наявних альтернатив. Для програміста вона надає шикарний API, що дозволяє використовувати протокол BitTorrent з надзвичайною легкістю, не замислюючись про зайві проблеми.

Перерахуємо переваги використання :

1. Спрощена процедура для створення/читання torrent – файлів. На швидкість роботи така універсальність і спрощеність не вплинули – вона знаходиться на висоті і всі дії (парсинг файлів, спілкування з сервером, переіндексація файлу) відбуваються дуже швидко.

2. Великий функціонал. Можливостей MN вистачить як для розробки клієнтських додатків (наприклад, Torrent – клієнтів), так і серверних (наприклад, Tracker – серверів). Всі необхідні класи вже реалізовані.

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

3. Простота використання. Код MonoTorrent написаний дуже якісно.

4. Крос-платформенність . МТ повністю крос-платформений .

Розглянемо вихідний код підключення та роботи з бібліотекою MonoTorrent .

```
// Шлях до папки , з якої ми працюємо
static string _programPath ;
// Папка, в яку будемо качати
static string _downloadPath ;
// Ім'я та шлях до файлу, який буде містити службову інформацію,
// необхідну для відновлення закачування
static string _fastResumeFile ;
// Шлях до торрент файлу.
static string _torrentPath ;
// Движок, який реалізує функції закачування
static ClientEngine _engine ;
// Допоміжний клас
static Top10Listener _listener ;
// Менеджер для зберігання закінчених налаштувань
// для чергового torrent - файлу
static TorrentManager _manager ;
```

Розглянемо функціонал скачування даних.

```
if ( args.Length < 2 )
{
Console.WriteLine ( " Введення параметрів: " ) ;
Console.WriteLine ( " <torrent path> <Download folder> " ) ;
Console.ReadKey ();
return ;
}
_programPath = Environment.CurrentDirectory ;
_torrentPath = args [0];
_downloadPath = args [ 1 ] ;
_fastResumeFile = _programPath + "\ temp.data " ;
_listener = new Top10Listener ( 10 ) ;
Console.CancelKeyPress +=
delegate {
exit ();
};
AppDomain.CurrentDomain.ProcessExit += delegate { exit (); } ;
AppDomain.CurrentDomain.UnhandledException +=
delegate(object sender, UnhandledExceptionEventArgs e )
```

						ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			41


```

Console.WriteLine ( " Автор : { 0 } " , _torrent.CreatedBy ) ;
Console.WriteLine ( " Створення : { 0 } " , _torrent.CreationDate ) ;
Console.WriteLine ( " Коментаріі: { 0 } " , _torrent.Comment ) ;
Console.WriteLine ( " Публіковано URL : { 0 } " , _torrent.PublisherUrl ) ;
Console.WriteLine ( " Розмір: { 0 } " , _torrent.Size ) ;
Console.WriteLine ( " Piece length : { 0 } " , _torrent.PieceLength ) ;
Console.WriteLine ( " Piece count : { 0 } " , _torrent.Pieces.Count ) ;
Console.WriteLine ( " Натисніть будь-яку клавішу для продовження ..." ) ;
Console.ReadKey () ;

if ( _fastResume.ContainsKey ( _torrent.InfoHash ) ) _manager =
new TorrentManager ( _torrent , _downloadPath , _torrentDef ,
new FastResume( ( BEncodedDictionary ) _fastResume[_torrent.InfoHash] ) ) ;
else
_manager=new TorrentManager(_torrent,_downloadPath,_torrentDef);
_engine.Register(_manager);
_manager.TorrentStateChanged + =
delegate(object o,TorrentStateChangedEventArgs e )
{
lock(_listener) _listener.WriteLine ( "Статус : " +
e.OldState.ToString ( )
+"Поточний статус: " + e.NewState.ToString());
} ;
foreach(TrackerTier ttier in _manager.TrackerManager.TrackerTiers )
{
foreach(MonoTorrent.Client.Tracker.Tracker tr in ttier.Trackers )
{
tr.AnnounceComplete+=delegate(object sender ,AnnounceResponseEventArgs e )
{
_listener.WriteLine(string.Format(" { 0 } : { 1 } " , e.Successful ,
e.Tracker.ToString ( ))) ;
};
}
}
_manager.Start();
int i=0 ;
bool _running = true ;
StringBuilder _stringBuilder = new StringBuilder ( 1024 ) ;
while ( _running )
{
if ( ( i + + ) % 10 == 0 )
{
if ( _manager.State == TorrentState.Stopped ) {

```

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

```

_running = false ;
exit ();
// Тут виводиться корисна інформація
// ( Швидкість закачування , кількість активних сполук тощо).
}
}
System.Threading.Thread.Sleep ( 500 ) ;

```

Розглянемо реалізацію ядра програми.

```

# Class:FileDropTarget
# Щоб вирішити перенесення з подальшим вивільненням
# для MY_torrent_client складають список в головному меню
class FileDropTarget(wx.FileDropTarget):
    def __init__(self, utility):
        # ініціалізуємо wsFileDropTarget Object
        wx.FileDropTarget.__init__(self)
        # Запам'ятовуємо значення об'єктів для пропущених файлів
        self.utility = utility
    def OnDropFiles(self, x, y, filenames):
        for filename in filenames:
            self.utility.queue.addtorrents.AddTorrentFromFile(filename)
        return True

# Class : MY_torrent_clientList
# MY_torrent_client список класів, які містять список торрентів
class MY_torrent_clientList(ManagedList):
    def __init__(self, parent):
        style = wx.LC_REPORT|wx.LC_VRULES|wx.CLIP_CHILDREN
        prefix = 'column'
        minid = 4
        maxid = 26
        exclude = []
        rightalign = [COL_PROGRESS,
                      COL_SIZE,
                      COL_DLSPEED,
                      COL_ULSPEED,
                      COL_RATIO,
                      COL_PEERPROGRESS,
                      COL_DLSize,
                      COL_ULSize,
                      COL_TOTALSPEED]

```

					БКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

```

ManagedList.__init__(self, parent, style, prefix, minid, maxid,
exclude, rightalign)
dragdroplist = FileDropTarget(self.utility)
self.SetDropTarget(dragdroplist)
self.lastcolumnsorted = -1
self.reversesort = False
self.Bind(wx.EVT_KEY_DOWN, self.OnKeyDown)
self.Bind(wx.EVT_LIST_COL_CLICK, self.OnColLeftClick)
self.Bind(wx.EVT_LIST_ITEM_RIGHT_CLICK, self.OnItemSelected)
# Подає додаткові деталі при лівому двійному кліку миші
self.Bind(wx.EVT_LEFT_DCLICK, self.OnLeftDClick)
# Надає локальні встановлення при натисканні середньої клавіші миші
self.Bind(wx.EVT_MIDDLE_DCLICK,
self.utility.actions[ACTION_LOCALUPLOAD].action)
# Робить дію, коли натиснута клавіша
def OnKeyDown(self, event):
    keycode = event.GetKeyCode()
    if event.CmdDown():
        if keycode == ord('a') or keycode == ord('A'):
# виділяє всі файли
            self.selectAll()
            elif keycode == ord('x') or keycode == ord('X'):
# інвертує обрані файли
                self.invertSelection()
            elif keycode == wx.WXK_RETURN or keycode ==
wx.WXK_NUMPAD_ENTER:
# відкриває додаткові деталі
                self.utility.actions[ACTION_DETAILS].action()
            elif keycode == wx.WXK_SPACE:
# відкриває локальні встановлення (Space)
                self.utility.actions[ACTION_LOCALUPLOAD].action()
            elif keycode == 399:
# відкриває меню при натисканні правої кнопки миші (windows menu key)
                self.OnItemSelected()
                event.Skip()
def OnColLeftClick(self, event):
    rank = event.GetColumn()
    colid = self.columns.getIDfromRank(rank)
    if colid == self.lastcolumnsorted:
        self.reversesort = not self.reversesort
    else:
        self.reversesort = False

```

```

self.lastcolumnsorted = colid
self.utility.queue.sortList(colid, self.reversesort)
def selectAll(self):
    self.updateSelected(select = range(0, self.GetItemCount()))
def updateSelected(self, unselect = None, select = None):
    if unselect is not None:
        for index in unselect:
            self.SetItemState(index, 0, wx.LIST_STATE_SELECTED)
    if select is not None:
        for index in select:
            self.Select(index)
self.SetFocus()
def getTorrentSelected(self, firstitemonly = False, reverse = False):
    queue = self.utility.queue
    torrentsselected = []
    for index in self.getSelected(firstitemonly, reverse):
        MY_torrent_clientTorrentTemp =
        queue.getMY_torrent_clientTorrent(index = index)
        if MY_torrent_clientTorrentTemp is not None:
            torrentsselected.append(MY_torrent_clientTorrentTemp)
    return torrentsselected
def OnItemSelected(self, event = None):
    selected = self.getTorrentSelected()
    if not selected:
        return
    popupmenu = MY_torrent_clientMenu(self.utility,
'menu_listrightclick')
# Випадаюче меню.
    if event is None:
# використовується позиція першого обраного пункту
# меню(подія при натисканні клавіши)
        MY_torrent_clientTorrentTemp = selected[0]
        position =
self.GetItemPosition(MY_torrent_clientTorrentTemp.listindex)
    else:
# використовується позиція курсора(подія при натисканні миші)
        position = event.GetPosition()
        self.PopupMenu(popupmenu, position)
def OnLeftDClick(self, event):
    event.Skip()
    self.utility.actions[ACTION_DETAILS].action()
# Class : MY_torrent_clientPanel

```

```

# Головний MY_torrent_client-клас панелей інтерфейсу користувача
class MY_torrent_clientPanel(wx.Panel):
    def __init__(self, parent):
        style = wx.CLIP_CHILDREN
        wx.Panel.__init__(self, parent, -1, style = style)
# Відладка вихідних даних.
    sys.stdout.write('Preparing GUI.\n');
    self.utility = parent.utility
    self.utility.window = self
    self.queue = self.utility.queue
# Список видалених подій торрентів
    self.postponedevents = []
#ручне додавання торрентів UI
    colSizer = wx.BoxSizer(wx.VERTICAL)
# Список клавiш управління UI
    self.list = MY_torrent_clientList(self)
    self.utility.list = self.list
    colSizer.Add(self.list, 1, wx.EXPAND|wx.ALL, 2)
    self.utility.bottomline2 = MY_torrent_clientBottomBar2(self)
    colSizer.Add(self.utility.bottomline2, 0, wx.ALL|wx.EXPAND, 3)
    self.SetSizer(colSizer)
    self.SetAutoLayout(True)
    self.list.SetFocus()
    def getSelectedList(self, event = None):
        return self.list
# Модифікація MY_torrent_client на льоту
    def updateColumns(self, force = False):
        # Модифікація показу у колонках для неактивних торрентів
        for MY_torrent_clientTorrentTemp in self.utility.torrents["all"]:
            MY_torrent_clientTorrentTemp.updateColumns(force = force)
# Class : MY_torrent_clientTaskBarIcon
# Іконка таскбара
class MY_torrent_clientTaskBarIcon(wx.TaskBarIcon):
    def __init__(self, parent):
        wx.TaskBarIcon.__init__(self)
        self.utility = parent.utility
        self.TBMENU_RESTORE = wx.NewId()
# встановлення іконки таскбара, та відстеження подій від нього
    self.Bind(wx.EVT_TASKBAR_LEFT_DCLICK, parent.onTaskBarActivate)
    self.Bind(wx.EVT_MENU, parent.onTaskBarActivate, id =
self.TBMENU_RESTORE)
    self.updateIcon()

```

```

def updateIcon(self):
    remove = True
    mintray = self.utility.config.Read('mintray', "int")
    if (mintray >= 2) or ((mintray >= 1) and
self.utility.frame.IsIconized()):
        remove = False
    if remove and self.IsIconInstalled():
        self.RemoveIcon()
    elif not remove and not self.IsIconInstalled():
        self.SetIcon(self.utility.icon, "MY_torrent_client")
def CreatePopupMenu(self):
    menu = wx.Menu()
    self.utility.actions[ACTION_STOPALL].addToMenu(menu, bindto = self)
    self.utility.actions[ACTION_UNSTOPALL].addToMenu(menu, bindto =
self)
    menu.AppendSeparator()
    menu.Append(self.TBMENU_RESTORE,
self.utility.lang.get('showMY_torrent_clientwindow'))
    self.utility.actions[ACTION_EXIT].addToMenu(menu, bindto = self)
    return menu

```

4.2 Захист розробленого програмного забезпечення

Захист розробленого програмного забезпечення буде відбуватися за допомогою алгоритму FEAL – блоковий шифр, запропонований Акіхіро Симідзу і Седзі Міягуті.

У ньому використовуються 64-бітовий блок і 64-бітовий ключ. Його ідея полягає і в тому, щоб створити алгоритм, подібний DES, але з більш сильною функцією етапу. Використовуючи менше етапів, цей алгоритм міг би працювати швидше. На жаль, дійсність виявилася далекою від цілей проекту.

Як вхід процесу шифрування використовується 64-бітовий блок відкритого тексту. Спочатку блок даних підлягає операції XOR з 64 бітами ключа. Потім блок даних розщеплюється на ліву і праву половини. Об'єднання лівої і правої половин за допомогою XOR утворює нову праву половину. Ліва половина і нова права половина проходять через N етапів (спочатку 4). На

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

кожному етапі половина об'єднується за допомогою функції $F[1]$ з 16 бітами ключа і за допомогою XOR – з лівою половиною, створюючи нову праву половину. Вихідна права половина (на початок етапу) стає новою лівою половиною. Після N етапів (ліва і права половини не переставляти після N -го етапу) ліва половина знову об'єднується з допомогою XOR з правою половиною, утворюючи нову праву половину, потім ліва і права об'єднуються разом в 64-бітове ціле. Блок даних об'єднується за допомогою XOR з іншими 64 бітами ключа і алгоритм завершується.

КБГПЗ-2023

					VKPM-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено головне вікно розробленої програми. Програма розроблена як класичний торрент-клієнт.

Тобто торрент-клієнт для мережного протоколу обміну файлами. У розробленій програмі файли розбиваються на невеликі частини й у такому вигляді передаються по мережі.

Торрент-клієнт закачує ці частини й потім збирає в себе файл воедино. У процесі завантаження шматочків, клієнт також віддає вже завантаженні частини, що дозволяє передавати торренти з великою швидкістю й без очікування звільнення джерела (сідера).

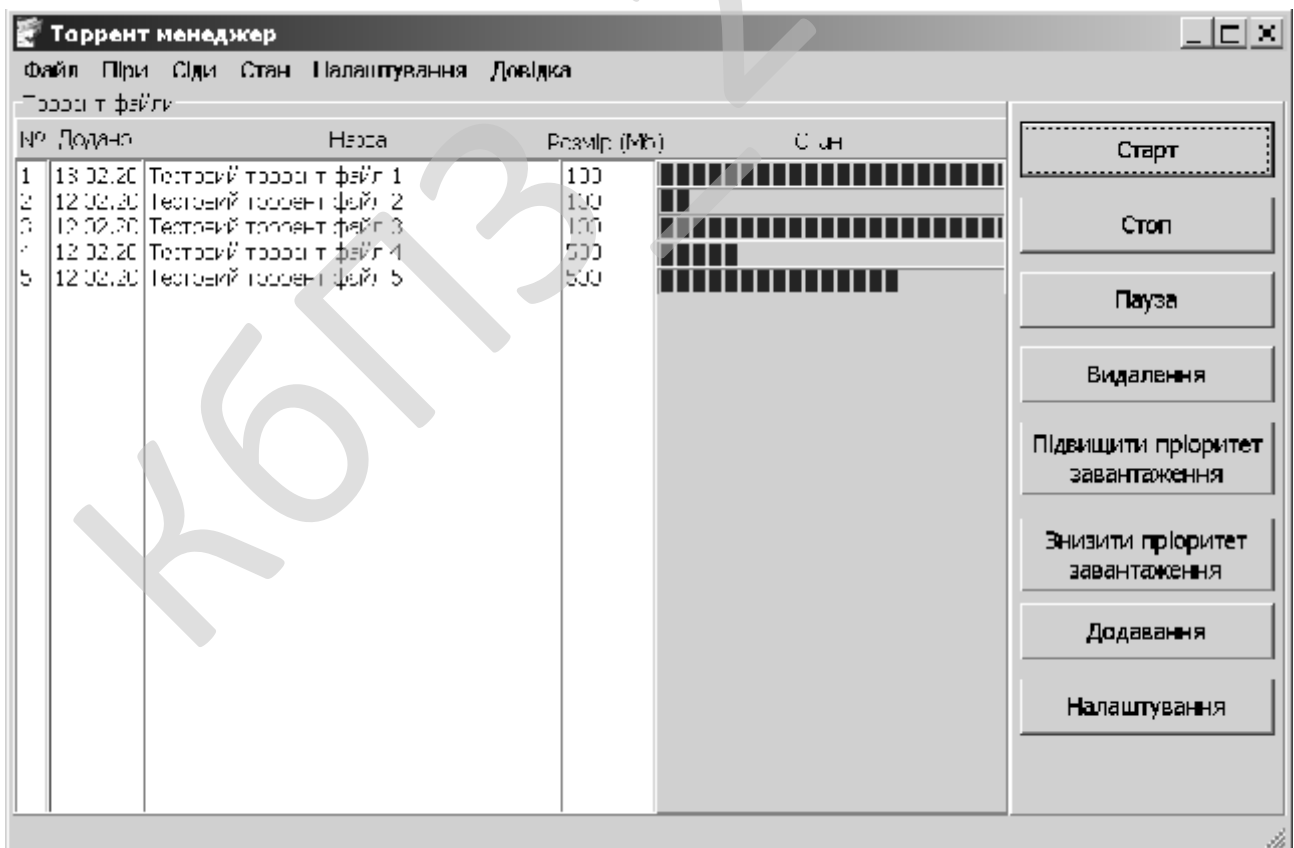


Рисунок 5.1 – Головне вікно програми

Для того щоб скачати торрент файл, клієнт з'єднується з торрент трекером, передає йому інформацію про свою IP адресу й геш суму файлу, який потрібно скачати.

Трекер відправляє клієтові IP адреси інших клієнтів, що також роздають або качають торрент. У процесі завантаження клієнт регулярно спілкується із сервером, повідомляючи інформацію про завантаження й одержуючи оновлений список IP адрес.

Клієнти передають інформацію прямо між собою без участі торрент-трекеру. Трекер тільки збирає дані із клієнтів про процес завантаження, підключених клієнтах і іншій інформації.

Для оптимальної роботи торрент протоколу потрібно, щоб максимальна кількість клієнтів могли приймати й віддавати файли.

При некоректному настроюванні міжмережного екрану/брандмауеру або трансляції адрес/NAT, швидкість передачі може значно зменшитися або припинитися зовсім.

На рисунку 5.2 зображено форму авторського права розробленого ПЗ.

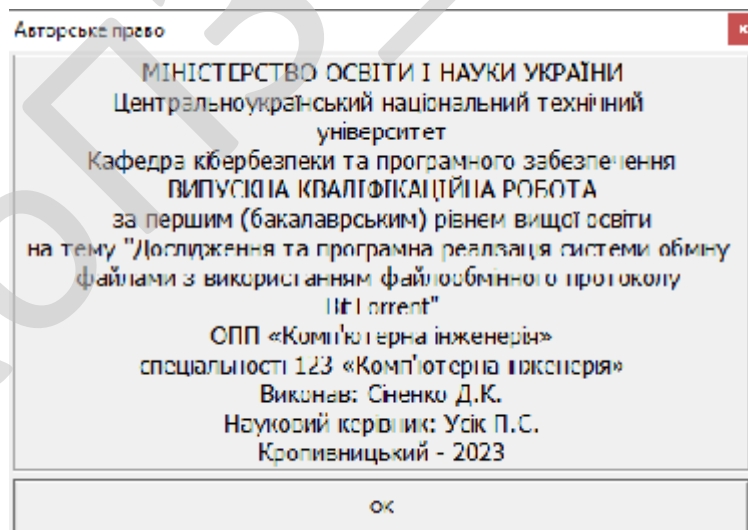


Рисунок 5.2 – Довідка автора

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи обміну файлами з використанням файлообмінного протоколу BitTorrent.

Метою розробки є дослідження та програмна реалізація системи обміну файлами з використанням файлообмінного протоколу BitTorrent.

Об'єктом дослідження є процес обміну файлами з використанням файлообмінного протоколу BitTorrent.

Предметом дослідження є методи обміну файлами з використанням файлообмінного протоколу BitTorrent.

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 обміну файлами з використанням файлообмінного протоколу BitTorrent.

– Розроблено вітчизняний продукт обміну файлами з використанням файлообмінного протоколу BitTorrent, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

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 обміну файлами з використанням файлообмінного протоколу BitTorrent. 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;
- г) 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	Frq	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, % :	Н _д	10
34. Норматив відрахувань у соціальні фонди, %	Н _с	22
35. Норматив загальногосподарських витрат, %	Н _г	15
36. Норматив витрат на освоєння нових мов програмування, %	Н _п	15
37. Рівень рентабельності програмної продукції, %	Р _е	50
38. Ставка податку на додану вартість, %	Н _{дв}	20

7.2 Розрахунок трудомісткості розробки програмної продукції

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

Визначимо трудомісткість розробки ПЗ для стадії РП.

Обчислюємо номінальні трудовитрати, люд-міс.:

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

де: A – коефіцієнт Боема, $A = 2,45$;

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

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.

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

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

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

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

де: C – визначений емпірично коефіцієнт, запропонований авторами методики, (МВ, додаток 4).

S – коефіцієнт стиснення (або подовження) графіка робіт %, що дозволяє коректувати терміни розробки ПЗ згідно встановленим вимогам. Вибираємо в межах (25...350)%. Приймаємо $S = 80$ %

$$T_{РП} = 0,3 \cdot 2,66 \cdot 9,37^{0,33 + 0,2(1,026 - 1,01)} \cdot 80 = 134 \text{ люд/день.}$$

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

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

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	134	Ф 7.1-7.4
Впровадження	13	Д13
Всього	175	–

7.3 Визначення чисельності виконавців і планового фонду зарплати

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

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

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

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

$$Ч = \frac{175 \cdot 1}{60 - 5} = 3,2 \text{ ставки.}$$

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

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

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

Найменування обладнання	Профілактичне обслуговування			
	Кількість хв. на один. обл.	Кількість обладнання	Затрати часу в хв.	Затрати часу в год.
Системний блок ПК	90	6	540	9
Монітор	60	6	360	6
Клавіатура	30	6	180	3
Маніпулятор «мишка»	30	6	180	3
Принтер матричний	60	0	0	0,0
Принтер лазерний	120	1	120	2
Принтер струминний	60	1	60	1
Сканер	20	1	20	0,33
Концентратор-маршрутизатор	30	1	30	0,5
Кабельні господарства ЛОМ на 1 м.п.	2,5	100	250	4,17
Копіювальний апарат	140	1	140	2,33
Усього за рік:			3 _ч	31,33

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:

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

$$\Phi_{\text{др}}^c = \frac{31 \cdot 3}{1,2} = 77,5 \text{ год.}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

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

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	14594	43782
Продакт-менеджер	0,25	14000	10500
Інженер-програміст	3,2	14500	139200
Інженер-електронщик	0,2	14000	8400
Інженер-системотехнік	0,25	14000	10500
Адміністратор мережі	0,5	14000	21000
Системний програміст	0,25	14000	10500
Дизайнер WEB	0,25	14000	10500
Інженер-верстальник	0,25	14000	10500
Бухгалтер-економіст	0,5	14000	21000
Всього за період розробки	$R_{cn} = 6,65$	-	$\Phi_{роб} = 285882$

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.

$$z_{cd} = \frac{285882}{6,65 \cdot 60} = 716,5 \text{ грн.}$$

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-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

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

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

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

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

Згідно даних інтернет ресурсу DOM.RIA (<https://dom.ria.com>) ціна одного квадратного метра площі, вік якої не перевищує 30 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:

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

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

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

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

Специфікація на обчислювальну техніку наведена в таблиці 7.7.

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

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

Продовження таблиці 7.6

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

Таблиця 7.7 – Балансова вартість обчислювальної техніки

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

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

Для визначення необхідної кількості капітальних вкладень складемо таблицю 7.8.

Таблиця 7.8 – Вартість основних фондів та амортизаційні відрахування розробника

Групи та види основних фондів	Балансова вартість, грн.	Амортизація	
		Норма, %	Відрахування, грн.
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 Daewoo Lanos SX 2010 взята по даним автобазару Авто-РІА, джерело https://auto.ria.com/uk/auto_daewoo_lanos_33568345.html, 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.

$$Z_o = 716,5 \cdot 175 / 19 = 6600 \text{ грн.}$$

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

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

де: H_q – норматив додаткової зарплати, %.

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

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

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

де: H_c – відрахування на соціальні потреби, %.

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

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

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

де: H_z – загальногосподарські витрати, %.

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

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

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

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

де: Z_{M1} – вартість паперу, грн.;

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

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

N_e – кількість екземплярів програм, шт.

Згідно прийнятих норм на підприємстві $n_{\text{вум}}$ приймаємо 0,5 пачки паперу на період розробки. Тоді, враховуючи, що вартість пачки паперу складає $U_n=210$ грн., визначаємо вартість паперу за період розробки:

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

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

Згідно прийнятих норм по комплектації до вартості запам'ятовуючих пристроїв входить вартість CD/DVD дисків. Їх кількість дорівнює кількості коробочних версій запропонованого продукту (приймаємо 5):

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

де: U_d – вартість дисків CD/DVD: CDR box – 21,4 грн./шт., DVD-R box – 35 грн./шт.

$$Z_{M2} = 5 \cdot 21,4 = 107 \text{ грн.}$$

Згідно норм одноразовій заправці підлягають усі друкуючі пристрої і становить:

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

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

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

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

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

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

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

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

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

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

Таблиця 7.9 – Нормативна калькуляція собівартості розробки програмного забезпечення задачі

Найменування статей витрат	Позначення	Величина, грн
1	2	3
1. Основна зарплата виконавців	Z_o	6600
2. Додаткова зарплата виконавців	Z_d	660
3. Відрахування на соціальні потреби	C_{oc}	2686
4. Загальногосподарські витрати	Γ_{ocn}	990
5. Витрати на матеріали	Z_M	101
6. Освоєння нових операційних систем, мов програмування	O_n	990
7. Амортизація основних фондів	A_m	2073
8. Повна собівартість програмного забезпечення	C_n	14100
9. Плановий прибуток	Π_p	7050
10. Ціна підприємства $C_n = C_n + \Pi_p$	C_n	21150
11. Податок на додану вартість $\Pi Д В = 0,01 \cdot H_{oc} \cdot C_n$	$\Pi Д В$	4230
12. Відпускна ціна програмної продукції $C = C_n + \Pi Д В$	C	25380

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

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

7.7 Визначення експлуатаційних витрат

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

Таблиця 7.11 – Розрахунок експлуатаційних витрат у споживача програмної продукції

Найменування статей витрат	Позначення	Сума витрат за варіантами, грн.	
		Базовий	Новий
1. Витрати на технічне обслуговування	Z_p	107360	21472
2. Витрати на електроенергію	$Z_{ел}$	20580	17640
3. Витрати на амортизацію	$Z_{ам}$	0	6345
Всього витрат за рік	I	127940	45457

Витрати на профілактичні роботи:

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

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

Z_2 – заробітна плата обслуговуючого персоналу, грн/год.

Після купівлі нового програмного забезпечення кількість профілактичних годин робіт зменшилася з 1000 годин на рік до 200 годин на рік, тому витрати на технічне обслуговування зменшилися з:

$$Z_{p \text{ баз}} = 1000 \cdot 80 \cdot 1,1 \cdot 1,22 = 107360 \text{ грн},$$

до:

$$Z_{p \text{ нов}} = 200 \cdot 80 \cdot 1,1 \cdot 1,22 = 21472 \text{ грн}.$$

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-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

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

$$Z_{ел баз} = 7 \cdot 0,35 \cdot 4000 \cdot 2,1 = 20580 \text{ грн.}$$

$$Z_{ел нов} = 7 \cdot 0,3 \cdot 4000 \cdot 2,1 = 17640 \text{ грн.}$$

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

Таблиця 7.12 – Розрахунок амортизаційних відрахувань

Групи основних фондів	Норма амортизації %	Балансова вартість, грн., за варіантами		Сума відрахувань, грн за варіантами	
		Базовий	Новий	Базовий	Новий
Програмна продукція	25	–	25380	–	6345
Всього відрахувань	-	–	25380	–	6345

7.8 Визначення економічної ефективності програмної продукції

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

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

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

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

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

Визначимо період окупності додаткових капітальних вкладень у виробника програмної продукції:

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

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.

$$E_{cn} = (127940 - 45457) - 0,25 \cdot 25380 = 76138 \text{ грн.}$$

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

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

$$T_{cn} = \frac{25380}{127940 - 45457} = 0,3 \text{ року.}$$

7.9 Висновки

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

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Характерною ознакою сучасного науково-технічного прогресу практично у всіх сферах діяльності людини є широке застосування комп'ютерних технологій, заснованих на використанні електронно-обчислювальних машин (ЕОМ). Сьогодні, а тим більше, майбутнє, вже важко уявити без комп'ютерів та іншої електронної техніки. Адже саме завдяки їм стала можливою швидка переробка величезних обсягів інформації, проведення необхідних розрахунків, виконання різних видів робіт, пов'язаних обробкою текстових та ілюстраційних зображень, організація оперативного отримання та передачі інформації, збереження її значних обсягів електронним способом.

Стрімке впровадження комп'ютерів не тільки в сфері управління виробництвом, в банківській системі, бізнесі, системі освіти, але також на транспорті, сфері обслуговування призвело до того, що десятки мільйонів людей у всьому світі виявились втягнутими у взаємодію людини з комп'ютером. Природно виникає запитання: настільки безпечною є ця взаємодія для людини? Адже відома аксіома про те, що будь-яка взаємодія людини та засобів праці двостороння.

Людина впливає на удосконалення засобів праці, а останні – на працюючу людину. Отже, навіть сучасні технології та техніка, до яких безперечно, залежать комп'ютерні технології та ЕОМ несуть у собі певні потенційні небезпеки. У зв'язку з цим набуває актуальності адекватна оцінка конкретних умов і характеру праці, яка сприяє обґрунтованому розробленню та впровадженню комплексу заходів і засобів, спрямованих на збереження здоров'я і працездатності людини в процесі праці за рахунок поліпшення параметрів виробничого середовища,

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

зменшення важкості, напруженості трудового процесу та збереження здоров'я працівників на комп'ютеризованих робочих місцях.

Законодавством України чітко врегульовано норми та вимоги до використання комп'ютерної техніки на підприємстві, безпосередньо й охорона праці на підприємстві при роботі за комп'ютером., зокрема «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями», затверджені наказом Мінсоцполітики від 14.02.2018 № 207 [1], «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98. [6].

Загальні вимоги пожежної безпеки під час експлуатації комп'ютерної техніки визначають «Правила пожежної безпеки в Україні» (затверджені наказом МВС від 30.12.2014 № 1417), комп'ютерних класів – пункт 3 розділу VIII «Правил пожежної безпеки для навчальних закладів та установ системи освіти України» (затверджені наказом МОН від 15.08.2016 № 974) та інші державні стандарти, що регламентують експлуатування комп'ютерної техніки як радіоелектронної апаратури.

8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

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

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

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

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

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

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

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

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

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

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

8.3 Розробка заходів з умов поліпшення охорони праці

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

- розміри 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-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

8.4 Пожежна безпека

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

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

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

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

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

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

Також встановлюється порядок роботи та технічного обслуговування:

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

Розробляються і впроваджуються правила роботи з відкритим вогнем і горючими матеріалами. Створюються графіки проходження інструктажів з пожежної безпеки співробітників, а також порядок і терміни перевірок знань

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

роботи з ЕОМ. Іноді може виникати необхідність написання допоміжних програм для поліпшення роботи вузла або для зниження витрат. З погляду забезпечення умов праці й вимог техніки безпеки для роботи програміста необхідно наступне: достатнє висвітлення екрана дисплея й робочого місця; повна технічна справність устаткування, його електробезпечність; достатня пожежна безпечність приміщення; оптимальний мікроклімат, що сприяє продуктивній роботі; відповідність робочого місця вимогам ергономіки. До небезпечних і шкідливих факторів, дії яких піддається програміст, можна віднести: можливість поразки електричним струмом, при електроні справності устаткування, порушенні заземлення або техніки безпеки; робота в мікрокліматі з неприпустимими параметрами; робота при недостатній освітленості екрана дисплея й робочого місця.

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

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

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

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		80

Необхідно передбачити наявність у межах досяжності первинних засобів гасіння пожежі (вогнегасників) для локалізації вогню власними засобами до приїзду команди пожежної охорони. Повинен бути розроблений план екстреної евакуації персоналу при виникненні загоряння. Кількість евакуаційних виходів повинне бути не менш двох. Допускається використання одного евакуаційного виходу, якщо відстань найбільш віддаленого робочого місця до цього виходу не перевищує 25 м.

8.5 Розрахунок занулення

Розрахунок занулення складається з трьох частин:

1. Розрахунок на відключаючу спроможність;
2. Визначення максимальної напруги на корпусі обладнання відносно землі при замиканні фази на корпус;
3. Розрахунок робочого і повторного заземлювачів

Початкові дані

1. Потужність електроприладів, які підлягають зануленню:

$$P = 8 \text{ кВт.}$$

2. Довжина магістрального кабеля: $L_M=32 \text{ м.}$

3. Довжина розгалуження (від розподільчого щита до електроприладів) :
 $l=15 \text{ м.}$

4. Матеріал провідників кабеля—алюміній.

5. Лінійна напруга $U=380 \text{ В.}$

6. Фазна напруга $U_f=220 \text{ В.}$

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

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

Центральноукраїн. нац. техн. ун-т. – 2-ге вид., перероб. та доп. – Кропивницький : ЦНТУ, 2019. – 27 с. URI: <http://dspace.kntu.kr.ua/jspui/handle/123456789/8769>

3. Закон України «Про охорону праці» від 14.10.1992 р. № 2694-ХІІ. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/2694-12> (дата звернення 19.10.23).

4. 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 19.09.23).

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

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

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи обміну файлами з використанням файлообмінного протоколу BitTorrent.

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 обміну файлами з використанням файлообмінного протоколу BitTorrent.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем обміну файлами з використанням файлообмінного протоколу BitTorrent.
- Досліджена система обміну файлами з використанням файлообмінного протоколу BitTorrent.
- На основі отриманих результатів досліджень створена програмна реалізація системи обміну файлами з використанням файлообмінного протоколу BitTorrent.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання обміну файлами з використанням файлообмінного протоколу BitTorrent.

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.

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

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.

Програма реалізована на мові високого рівня Python. 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 FEAL.

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 76138 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 0,3 роки.

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Сіненко Д.К. Дослідження та програмна реалізація системи обміну файлами з використанням файлообмінного протоколу BitTorrent // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2023.
2. Ramon Nastase «Computer Networking: The Beginner’s guide for Mastering Computer Networking, the Internet and the OSI Model». 2018. – 186 p.
3. Russ White & Ethan Banks «Computer Networking Problems and Solutions: An Innovative Approach to Building Resilient, Modern Networks». 2017. – 832 p.
4. 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.
5. Smirnova, T., Gnatyuk, S., Yudin, O., Sydorenko, V., Polozhentsev, A., «The Model for Calculating the Quantitative Criteria for Assessing the Security Level of Information and Telecommunication Systems». *CEUR Workshop Proceedings Volume 3156*, 2022, Pages 390-399.
6. Smirnova T., Gnatyuk S., Berdibayev R., Avkurova Zh., Iavich M. «Cloud-Based Cyber Incidents Response System and Software Tools». *Communications in Computer and Information Science*, 2021, vol 1486. Springer, Cham. pp 169-184.
7. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». *CEUR Workshop Proceedings*. Volume 2740, 2020, Pages 102-114.
8. 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.

9. 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.

10. 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.

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

12. 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.

13. 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.

14. 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.

15. 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.

16. 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.

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

18. 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.

19. 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.

20. Smirnov, O., Kuznetsov, A., Kiian, A., Zamula, A., Rudenko, S., Hryhorenko, V., «Variance Analysis of Networks Traffic for Intrusion Detection in Smart Grids», *2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS)*, Kyiv, Ukraine April 17-19, 2019 P. 353-358.

21. 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.

22. 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.

23. 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.

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

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

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

27. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Поліщук Л.І., Смірнов С.А. «Дослідження статистичної стійкості та швидкісних характеристик запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Вісник Хмельницького національного університету. Серія: «Технічні науки»*, № 2 (307). С. 46-52. 2022.

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

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		89

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

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

31. 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.

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

33. 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.

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

35. 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.

36. 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

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

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).

37. 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

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

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

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

41. Смірнов О.А., Гнатюк С.О., Кавун С.В., Терейковський І.А., Жмурко Т.О., Смірнов С.А., Коваленко А.С. Основи безпеки в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2018. – 177 с.

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

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

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

44. 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.

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

46. Смірнов О.А., Смірнов С.А., Дідик А.К. Метод безпечної маршрутизації метаданих у хмарні антивірусні системи. Системи озброєння та військова техніка. – Випуск 2 (46) – Х.: ХУПС – 2016. – С. 146-149.

47. Смірнов О.А., Кавун С.В., Доренський О.П., Вялкова В.І. Інформаційна безпека в комп'ютерних мережах. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 151 с.

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

49. 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.

50. Смірнов О.А., Стасєв Ю.В., Бараннік В.В. Захист інформації в автоматизованих системах управління. Навчальний посібник – Харків: ХУПС, 2015. – 264 с.

51. Смірнов О.А., Мелешко Є.В., Семенов С.Г. Методи та засоби обробки сигналів і даних в інформаційних системах. Навчальний посібник для студентів вищих навчальних закладів напрямів підготовки 8.050102 «Комп'ютерна інженерія». За ред. О.А. Смірнова Гриф «Навчальний посібник» надано у відповідності з листом Міністерства освіти і науки, молоді та спорту України від 17.04.2012 року № 1/11-5249. – Кіровоград: КНТУ 2012. – 250 с.

					ВКРМ-123.23.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

Додаток А
(обов'язковий)

Технічне завдання

Зміст

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

					ВКРМ-123.23.0019.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Сіненко Д.К.				<i>Дослідження та програмна реалізація системи обміну файлами з використанням файлообмінного протоколу BitTorrent</i>	Літ.	Аркуш	Аркушів
Перевірів	Усік П.С.					М	1	6
Н. Контр.	Коваленко А.С.				ЦНТУ КІ-22М-1			
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи обміну файлами з використанням файлообмінного протоколу BitTorrent.

2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 34-13 від 04.08.2023 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи обміну файлами з використанням файлообмінного протоколу BitTorrent.

4 Джерела розробки

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

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

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

					ВКРМ-123.23.0019.00.00.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 обміну файлами з використанням файлообмінного протоколу BitTorrent;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

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

5.4 Вимоги до архітектури

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

5.5 Вимоги до надійності

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

					ВКРМ-123.23.0019.00.00.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 Python.

					ВКРМ-123.23.0019.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

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

7 Економічні вимоги

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

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

8 Вимоги щодо охорони праці

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

					ВКРМ-123.23.0019.00.00.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 – 92 аркуші.

10 Етапи розробки

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

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

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

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Розрахунок з техніко-економічного обґрунтування.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

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

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

					ВКРМ-123.23.0019.00.00.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
_____ Усік П.С.

*Дослідження та програмна реалізація
системи обміну файлами з використанням файлообмінного протоколу
BitTorrent*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 78

Літера: РП

CH_BitTorrent_client_engine.py - двигун клієнта торренту

```

import wx
import sys

from operator import itemgetter
from threading import Event, Timer
from time import time
#from cStringIO import StringIO
#from traceback import print_exc

from BitTornado.clock import clock

from binascii import unhexlify

from BitTornado.download_bt1 import BT1Download

from Utility.constants import * #IGNORE:W0611
from Utility.helpers import getfreespace

#####
#
# Class: CH_BitTorrent_client_Engine
#
# Двигун торренту.
#
#####
class CH_BitTorrent_client_Engine:
    def __init__(self, BitTorrent, myid):
        self.BitTorrent = BitTorrent
        self.queue = BitTorrent.queue
        self.utility = BitTorrent.utility

        self.downsize = { 'old' : BitTorrent.files.downsize,
                          'new' : 0.0 }
        self.upsized = { 'old' : BitTorrent.files.upsized,
                         'new' : 0.0 }

        self.spewwait = time()

        self.reannouncelast = 0
        self.lastexternalannounce = ''

        self.timers = { 'lastupload': time(),
                       'lastdownload': time(),
                       'infrequent': None }

        self.btstatus = self.utility.lang.get('waiting')
        self.progress = BitTorrent.files.progress

        self.eta = None
        self.rate = { "down" : 0.0,
                      "up" : 0.0 }

        self.numpeers = 0
        self.numseeds = 0

        self.numconnections = 0
        self.numcopies = None
        self.peeravg = None
        self.totalspeed = 0.0
        self.numerrortracker = 0
        self.lasterrortracker = 0

        self.current_ratesetting = None

        self.seedingtimelastcheck = None

```

```

# Обчислюємо mean над декількома останніми значеннями uprate/downrate
self.pastrate = { "down" : [0.0] * 20,
                  "up" : [0.0] * 20 }
self.meanrate = { "down" : 0.0,
                  "up" : 0.0 }

# Створюємо потік, якщо ми під'єднались
self.hasConnections = False

self.doneflag = Event()

# Починаємо закачку ....

self.controller = self.utility.controller

self.waiting = True
self.checking = False
self.working = False
self.seed = False
self.closed = False

self.status_err = ['']
self.status_errtime = 0
self.status_done = 0.0

self.response = self.BitTorrent.metainfo
hash = unhexlify(self.BitTorrent.infohash)

self.rawserver = self.controller.handler.newRawServer(hash,
self.doneflag)

self.workarounds = { 'hasexternal': False }

self.color = 'color_startup'

btconfig = self.utility.getBTParams()

#     if self.BitTorrent.files.getSpaceNeeded(realsize = False) == 0:
#         if self.BitTorrent.files.skipcheck or \
#             (self.BitTorrent.status.completed and
self.utility.config.Read('skipcheck', "boolean")):
#             btconfig['check_hashes'] = 0
#             self.BitTorrent.files.skipcheck = False

self.dow = BTDownload(self.display,
                      self.finished,
                      self.error,
                      self.controller.exchandler,
                      self.doneflag,
                      btconfig,
                      self.response,
                      hash,
                      myid,
                      self.rawserver,
                      self.controller.listen_port)

#####
# ПОЧИНАЄМО ЗАКАЧКУ (для одного порту)
#####

def start(self):
    # Видаляємо інформацію з кешу
    if ((self.BitTorrent.status.value == STATUS_HASHCHECK) and
        self.utility.config.Read('fastresume', "boolean")):
        self.dow.appdataobj.deleteBitTorrentData(self.dow.infohash)

```

```

if not self.dow.saveAs(self.saveAs):
    self.shutdown()
    return
self._hashcheckfunc = self.dow.initFiles()

if not self._hashcheckfunc:
    self.shutdown()
    return

self.controller.hashchecksched(self.BitTorrent)

def hashcheck_start(self, donefunc):
    if self.is_dead():
        self.shutdown()
        return
    self.waiting = False
    self.checking = True

    # Запускаємо визначені завдання
    self.InfrequentTasks()

    self._hashcheckfunc(donefunc)

def hashcheck_callback(self):
    self.checking = False

    if self.is_dead():
        self.shutdown()
        return

    if not self.dow.startEngine(ratelimiter = None):
        self.shutdown()
        return

    self.dow.startRerequester()
    self.statsfunc = self.dow.startStats()

    if self.BitTorrent.status.value != STATUS_HASHCHECK:
        self.BitTorrent.status.updateStatus(STATUS_ACTIVE)

    self.BitTorrent.files.setFilePriorities()
    self.BitTorrent.connection.setMaxInitiate()

    # Встановлюємо прапорець
    if self.BitTorrent.dialogs.details is not None:
        self.dow.spewflag.set()

    self.dow.spewflag.set()

    self.rawserver.start_listening(self.dow.getPortHandler())
    self.working = True

def shutdown(self):
    # Видаляємо зі списку активних торрентів
    try:
        del self.utility.BitTorrents["active"][self.BitTorrent]
    except:
        pass

    # Відключаємо відображення часу зачачування
    try:
        if self.timers['infrequent'] is not None:
            self.timers['infrequent'].cancel()
    except:
        pass

    if self.closed:
        return
    self.doneflag.set()

```

```

try:
    self.rawserver.shutdown()
except:
    pass

try:
    self.dow.shutdown()
except:
    pass
self.waiting = False
self.checking = False
self.working = False
self.closed = True
self.controller.was_stopped(self.BitTorrent)

def display(self, activity = None, fractionDone = None):
    # реально використовується тільки StorageWrapper
    self.setActivity(activity)
    if fractionDone is not None:
        self.status_done = float(fractionDone)

def error(self, msg):
    if self.doneflag.isSet():
        self.shutdown()
    self.status_err.append(msg)
    self.status_errtime = clock()
    self.errormsg(msg)

def saveAs(self, name, length, saveas, isdir):
    return self.BitTorrent.files.dest

def done(self, event = None):
    self.BitTorrent.set()

def is_dead(self):
    return self.doneflag.isSet()

#####
# Закінчення ПОЧИНАЄМО ЗАКАЧКУ (ДЛЯ ОДНОГО ПОРТУ)
#####

def setActivity(self, activity):
    if activity is not None:
        activities = { "checking existing data":
self.utility.lang.get('checkingdata'),
                    "allocating disk space":
self.utility.lang.get('allocatingspace'),
                    "moving data": self.utility.lang.get('movingdata') }
        self.btstatus = activities.get(activity, activity)

    def onUpdateStatus(self, fractionDone = None, timeEst = None, downRate =
None, upRate = None, activity = None, statistics = None, spew = None):

        # На всяк випадок завершуємо торрент
        #
        self.BitTorrent.status.completed = self.seed

        # оновлюємо дату кожні 20 хвилин
        #####
        if self.utility.config.Read('scrape', "boolean"):
            self.BitTorrent.actions.scrape()

        # виводимо дату на екран
        #####
        if fractionDone is not None and not self.seed:
            self.progress = (float(fractionDone) * 100)

        self.setActivity(activity)

```

```

if timeEst is not None:
    self.eta = timeEst
else:
    self.eta = None

if self.BitTorrent.status.value != STATUS_PAUSE:
    if not self.seed and downRate is not None:
        self.rate['down'] = float(downRate)
        if self.rate['down'] != 0.0:
            self.timers['lastdownload'] = time()
    else:
        self.rate['down'] = 0.0

    if upRate is not None:
        self.rate['up'] = float(upRate)

        self.pastrate['up'].append(self.rate['up'])
        self.pastrate['up'].pop(0)

        if self.rate['up'] != 0.0:
            self.timers['lastupload'] = time()

        #

        total = sum(self.pastrate['up'])
        self.meanrate['up'] = total / 20
    else:
        self.rate['up'] = 0.0

if statistics is not None:
    self.numpeers = statistics.numPeers
    self.numcopies = statistics.numCopies
    self.peeravg = statistics.percentDone

    # Модифікація завантаження, кількості інформації, яку ще треба
    # завантажити, та прогресу завантаження
    self.downsize['new'] = float(statistics.downTotal)
    self.upsized['new'] = float(statistics.upTotal)
    self.BitTorrent.files.updateProgress()
    self.totalspeed = float(statistics.BitTorrentRate)
    self.numconnections = statistics.numPeers

    if not self.seed:
        self.numseeds = statistics.numSeeds
        self.numconnections += statistics.numSeeds
    else:
        self.numseeds = statistics.numOldSeeds
else:
    self.peeravg = None
    self.numcopies = None

# Модифікація кольору
self.updateColor(statistics, spew)

if self.seed:
    self.countSeedingTime()
    if self.BitTorrent.status.isDoneUploading():
        self.TerminateUpload()

# Модифікація текстових рядків
self.BitTorrent.updateColumns([COL_PROGRESS,
                                COL_BTSTATUS,
                                COL_ETA,
                                COL_DLSPEED,
                                COL_ULSPEED,
                                COL_MESSAGE])

if statistics is not None:
    # Співвідношення завантаженості, #Сіди, #Піри, #Копіювання, #Піри
    # відображення завантаження,

```

```

# Розмір закачаних даних, Розмір даних, які ще треба закачати,
Загальна швидкість
self.BitTorrent.updateColumns([COL_RATIO,
                                COL_SEEDS,
                                COL_PEERS,
                                COL_COPIES,
                                COL_PEERPROGRESS,
                                COL_DLSIZE,
                                COL_ULSIZE,
                                COL_TOTALSPEED,
                                COL_SEEDTIME,
                                COL_CONNECTIONS])

# Модифікація відображення прогресу завантаження
if statistics is not None:
    self.BitTorrent.files.updateFileProgress(statistics)

try:
    self.updateDetailWindow(statistics, spew)
except:
    # На всяк випадок встановлюємо вікно у недоступне
    # або знищуємо першу закачку
    pass

if self.BitTorrent.status.value == STATUS_HASHCHECK and self.working:
    # Запуск процедури, якщо торрент активний, після перевірки хеш-
функції файлів
    activevalues = [ STATUS_ACTIVE, STATUS_PAUSE, STATUS_SUPERSEED ]
    oldstatus = self.BitTorrent.actions.oldstatus
    if not oldstatus in activevalues:
        self.shutdown()
        return
    else:
        self.BitTorrent.status.updateStatus(STATUS_ACTIVE)

def getExternalConnectionsMade(self, spew):
    if self.workarounds['hasexternal']:
        return True

    # видаляємо одне з'єднання
    for x in range(len(spew)):
        if spew[x]['direction'] == 'R':
            self.workarounds['hasexternal'] = True

    # Очищуємо прапорці...
    if self.BitTorrent.dialogs.details is None:
        self.dow.spewflag.clear()
        return True

    return False

def updateColor(self, statistics = None, spew = None):
    #####
    # Настроювання кольору :
    #####
    color = None

    if statistics is not None:
        externalConnectionMade = statistics.external_connection_made

        if externalConnectionMade:
            self.workarounds['hasexternal'] = True
        else:
            externalConnectionMade = self.workarounds['hasexternal']
            if not externalConnectionMade and spew is not None:
                externalConnectionMade =
self.getExternalConnectionsMade(spew)

    if statistics is None:

```

```

        color = 'color_startup' #Start up
        self.hasConnections = False
    elif statistics.numPeers + statistics.numSeeds + statistics.numOldSeeds
    == 0:
        if statistics.last_failed:
            #Роз'єднання
            color = 'color_disconnected'
        else:
            #Немає з'єднання
            color = 'color_noconnections'
            self.hasConnections = False
    elif (not externalConnectionMade):
        #Немає поступаючи потоків
        color = 'color_noincoming'
        self.hasConnections = True
    elif ((statistics.numSeeds + statistics.numOldSeeds == 0)
          and ((self.seed and statistics.numCopies < 1)
              or (not self.seed and statistics.numCopies2 < 1))):
        #Не повністю завантажений файл
        color = 'color_nocomplete'
        self.hasConnections = True
    else:
        #Все добре
        color = 'color_good'
        self.hasConnections = True

    self.color = color
    self.BitTorrent.updateColor()

#
# Необхідно зробити на кожному проході через updateStatus
#
def InfrequentTasks(self):
    try:
        if self.timers['infrequent'] is not None:
            self.timers['infrequent'].cancel()
    except:
        pass

    self.CheckTimeouts()
    self.CheckDiskSpace()

    # Повинен перевірити дисковий простір частіше, поки в стадії
    «Розміщуючи»
    if self.waiting or self.checking:
        nextcheck = 2
    else:
        nextcheck = 30

    self.timers['infrequent'] = Timer(nextcheck, self.InfrequentTasks)
    self.timers['infrequent'].start()

#
# Перевіримо, щоб переконатися, що є вільний дисковий простір
#
def CheckDiskSpace(self):
    threshold = self.utility.config.Read('diskfullthreshold', "int")

    # Disk checking is disabled
    if threshold == 0:
        return

    # Дивимося скільки потрібно простору для завантаження торренту
    spaceneeded = self.BitTorrent.files.getSpaceNeeded()

#
if spaceneeded == 0L:
    return

```

```

dest = self.BitTorrent.files.getProcDest(pathonly = True)
if dest is None:
    #
    return

spaceleft = getfreespace(dest)
if spaceleft < long((2**20) * threshold):
    message = self.utility.lang.get('diskfull') + \
        " (" + self.utility.size_format(spaceleft) + ")"
    self.errormsg(message)
    self.utility.actionhandler.procSTOP([self.BitTorrent])

#
# Дивимося, скільки необхідно часу для завантаження
#
def CheckTimeouts(self):
    # Перевіряємо, щоб поглянути якщо нам необхідно перевірити час
    if not self.BitTorrent.connection.timeout or
self.BitTorrent.status.value == STATUS_PAUSE:
        return

    # Не перевіряємо жодної передачі завантаження за 30 хв.
    # (коли торрент є лічем)
    #####
    if not self.seed:
        timeoutdownload = self.utility.config.Read('timeoutdownload')
        if (timeoutdownload != 'oo'
            and (time() - self.timers['lastdownload']) >
(float(timeoutdownload)*60)):
            self.ReducePrioandForceQueue()
            return

    # Не перевіряємо жодної передач назовні за 1 годину
    # (коли торрент є сидом)
    #####
    else:
        timeoutupload = self.utility.config.Read('timeoutupload')
        if ((timeoutupload != 'oo')
            and (time() - self.timers['lastupload']) >
(float(timeoutupload)*3600)):
            self.ReducePrioandForceQueue()
            return

def updateDetailWindow(self, statistics = None, spew = None):
    #####
    # Деталізуємо вид вікна клієнта торрента
    #####
    detailwin = self.BitTorrent.dialogs.details
    if detailwin is None or not detailwin.update:
        return
    detailpanel = detailwin.detailPanel

    if statistics is not None:
        detailpanel.updateFromCH_BitTorrent_clientBitTorrent()

    if spew is not None and (time() - self.spewwait > 1):
        self.updateSpewList(statistics, spew)

    if statistics is not None:
        detailpanel.storagestats1.SetLabel("          " +
self.utility.lang.get('detailline1')
            % (statistics.storage_active,
                statistics.storage_new,
                statistics.storage_dirty))
        detailpanel.storagestats2.SetLabel("          "+
self.utility.lang.get('detailline2')
            % (statistics.storage_numcomplete,
                statistics.storage_totalpieces,
                statistics.storage_justdownloaded,

```

```

statistics.storage_numflunked))

def updateSpewList(self, statistics = None, spew = None):
    detailwin = self.BitTorrent.dialogs.details
    if spew is None or detailwin is None or not detailwin.update:
        return

    self.spewwait = time()
    spewList = detailwin.detailPanel.spewList
    columns = spewList.columns

    numcols = len(columns.active)
    # (no point in doing anything if there aren't any columns to update)
    if numcols == 0:
        return

    spewlen = len(spew) + 3

    if statistics is not None:
        kickbanlen =
len(statistics.peers_kicked)+len(statistics.peers_banned)
        if kickbanlen:
            spewlen += kickbanlen + 1
        if statistics.peers_kicked:
            spewlen += 1
        if statistics.peers_banned:
            spewlen += 1
    else:
        kickbanlen = 0

    try:
        for x in range(spewlen-spewList.GetItemCount()):
            i = wx.ListItem()
            spewList.InsertItem(i)
        for x in range(spewlen, spewList.GetItemCount()):
            spewList.DeleteItem(len(spew) + 1)
    except wx.PyDeadObjectError:
        pass

    tot_uprate = 0.0
    tot_downrate = 0.0

    # Sort by uprate first
    spew.sort(key = itemgetter('uprate'), reverse = True)
    if not self.BitTorrent.status.completed:
        # Then sort by downrate if not complete
        spew.sort(key = itemgetter('downrate'), reverse = True)

    for x in range(len(spew)):
        for colid, rank in columns.active:
            self.updateSpewColumnText(x, rank, self.getSpewColumnText(colid,
x, spew))

            tot_uprate += spew[x]['uprate']
            tot_downrate += spew[x]['downrate']

    x = len(spew)
    for i in range(numcols):
        self.updateSpewColumnText(x, i, '')

    x += 1
    for colid, rank in columns.active:
        if colid == SPEW_IP:
            text = self.utility.lang.get('TOTALS')
        else:
            text = ''
        self.updateSpewColumnText(x, rank, text)

```

```

x += 1
for colid, rank in columns.active:
    if colid == SPEW_UP:
        text = self.utility.speed_format(tot_uprate, truncate = 0)
    elif colid == SPEW_DOWN:
        text = self.utility.speed_format(tot_downrate, truncate = 0)
    elif colid == SPEW_DLSize and statistics is not None:
        text = self.utility.size_format(float(statistics.downTotal))
    elif colid == SPEW_ULSize and statistics is not None:
        text = self.utility.size_format(float(statistics.upTotal))
    else:
        text = ''
    self.updateSpewColumnText(x, rank, text)

if kickbanlen:
    x += 1
    for i in range(numcols):
        self.updateSpewColumnText(x, i, '')

    if statistics.peers_kicked:
        x += 1
        for colid, rank in columns.active:
            if colid == SPEW_IP:
                text = self.utility.lang.get('KICKED')
            else:
                text = ''
            self.updateSpewColumnText(x, rank, text)

        for ip in statistics.peers_kicked:
            x += 1
            for colid, rank in columns.active:
                if colid == SPEW_IP:
                    text = ip[1]
                else:
                    text = ''
                self.updateSpewColumnText(x, rank, text)

    if statistics.peers_banned:
        x += 1
        for colid, rank in columns.active:
            if colid == SPEW_IP:
                text = self.utility.lang.get('BANNED')
            else:
                text = ''
            self.updateSpewColumnText(x, rank, text)

        for ip in statistics.peers_banned:
            x += 1
            for colid, rank in columns.active:
                if colid == SPEW_IP:
                    text = ip[1]
                else:
                    text = ''
                self.updateSpewColumnText(x, rank, text)

def updateSpewColumnText(self, line, colid, text):
    detailwin = self.BitTorrent.dialogs.details
    if detailwin is None or not detailwin.update:
        return

    try:
        detailwin.detailPanel.spewList.SetStringItem(line, colid, text)
    except wx.PyDeadObjectError:
        pass

def getSpewColumnText(self, colid, line, spew):
    text = None

```

```

starflag = { True : '*', False : ' ' }

if colid == SPEW_UNCHOKE:
    text = starflag[spew[line]['optimistic']]

elif colid == SPEW_IP:
    text = spew[line]['ip']

elif colid == SPEW_LR:
    if spew[line]['direction'] == 'R':
        text = self.utility.lang.get('spew_direction_remote')
    else:
        text = self.utility.lang.get('spew_direction_local')

elif colid == SPEW_UP:
    if spew[line]['uprate'] > 100:
        text = self.utility.speed_format(spew[line]['uprate'], truncate
= 0, stopearly = "KB")

elif colid == SPEW_INTERESTED:
    text = starflag[spew[line]['uinterested']]

elif colid == SPEW_CHOKING:
    text = starflag[spew[line]['uchoked']]

elif colid == SPEW_DOWN:
    if spew[line]['downrate'] > 100:
        text = self.utility.speed_format(spew[line]['downrate'],
truncate = 0, stopearly = "KB")

elif colid == SPEW_INTERESTING:
    text = starflag[spew[line]['dinterested']]

elif colid == SPEW_CHOKED:
    text = starflag[spew[line]['dchoked']]

elif colid == SPEW_SNUBBED:
    text = starflag[spew[line]['snubbed']]

elif colid == SPEW_DLSize:
    text = self.utility.size_format(float(spew[line]['dtotal']))

elif colid == SPEW_ULSize:
    if spew[line]['utotal'] is not None:
        text = self.utility.size_format(float(spew[line]['utotal']))

elif colid == SPEW_PEERPROGRESS:
    text = '%.1f%' % (float(int(spew[line]['completed']*1000))/10)

elif colid == SPEW_PEESSPEED:
    if spew[line]['speed'] is not None:
        text = self.utility.speed_format(spew[line]['speed'], truncate =
0)

if text is None:
    text = ""

return text

def errormsg(self, errormsg):
    errors = {"проблеми при з'єднанні з трекером":
self.utility.lang.get('trackererror_problemconnecting'),
"знехтувано трекером":
self.utility.lang.get('trackererror_rejected'),
"погані дані з трекеру":
self.utility.lang.get('trackererror_baddata') }

try:
    trackererror = False

```

```

for error in errors:
    index = errormsg.lower().find(error)
    if index != -1:
        oldlen = len(error)
        errormsg = errormsg[:index] + errors[error] + errormsg[index
+ oldlen:]

        trackererror = True
        break
if trackererror:
    currenttime = time()
    if self.lasterrortracker == 0:
        self.lasterrortracker = currenttime

    if (currenttime - self.lasterrortracker) < 120: #error with in 2
mins

        self.numerrortracker += 1
    else:
        self.numerrortracker = 0
        self.lasterrortracker = currenttime
except:
    pass

self.BitTorrent.changeMessage(errormsg, "error")

# якщо помилка з'єднання з трекером в параметрі 'timeouttracker'
вказуємо кількість хвилин
#####
if self.BitTorrent.connection.timeout and
self.utility.config.Read('timeouttracker') != "oo":
    try:
        if self.numerrortracker >
self.utility.config.Read('timeouttracker', "int"):
            self.ReducePrioandForceQueue()
    except:
        pass

def ReducePrioandForceQueue(self):
    currentprio = self.BitTorrent.prio
    if currentprio < 4: #пріоритет не нижчий
        self.BitTorrent.changePriority(currentprio + 1) #нижчий першого
пріоритету

    self.queueMe()

def countSeedingTime(self):
    now = time()
    if self.seedingtimelastcheck is None:
        lastcheck = now
    else:
        lastcheck = self.seedingtimelastcheck
    timelapse = now - lastcheck

    self.BitTorrent.connection.seedingtime += timelapse

    if self.BitTorrent.connection.getSeedOption('uploadoption') == "1":
        self.BitTorrent.connection.seedingtimeleft =
self.BitTorrent.connection.getTargetSeedingTime() -
self.BitTorrent.connection.seedingtime
    elif self.BitTorrent.connection.getSeedOption('uploadoption') == "2":
        if self.meanrate['up'] > 0:
            if self.BitTorrent.files.downsize == 0.0 :
                down = self.BitTorrent.files.floattotalsize
            else:
                down = self.BitTorrent.files.downsize
            up = self.BitTorrent.files.upsized
            ratio =
float(self.BitTorrent.connection.getSeedOption('uploadratio'))
            required = ((ratio / 100.0) * down) - up

```

```

        newseedingtimeleft = required / self.meanrate['up']
        delta = max(newseedingtimeleft/10, 2)
        if abs(self.BitTorrent.connection.seedingtimeleft -
newseedingtimeleft) > delta:
            # If timer value deviates from theoretical value by more
            # then 10%, reset it to theoretical value
            self.BitTorrent.connection.seedingtimeleft =
newseedingtimeleft
        else:
            # Keep on timing
            self.BitTorrent.connection.seedingtimeleft -= timelapse
            if self.BitTorrent.connection.seedingtimeleft < 0.1:
                self.BitTorrent.connection.seedingtimeleft = 0.1
        else:
            # Set to 366 days (= infinite)
            self.BitTorrent.connection.seedingtimeleft = 9999999999999999

self.seedingtimelastcheck = now

def TerminateUpload(self):
    # Закінчення прогресу завантаження
    self.BitTorrent.status.completed = True
    self.progress = 100.0

    self.BitTorrent.connection.stopEngine()

    self.queue.updateAndInvoke()

def finished(self):
    self.seed = True

    # процес сіда
    self.BitTorrent.status.completed = True
    self.progress = 100.0
    self.BitTorrent.files.updateProgress()

    if self.BitTorrent.status.isDoneUploading():
        self.TerminateUpload()

    # Модифікація колонок 5, 6, 8, 10
    self.BitTorrent.updateColumns([COL_PROGRESS,
                                   COL_BTSTATUS,
                                   COL_ETA,
                                   COL_DLSPEED])

    self.BitTorrent.updateColor()

    self.queue.updateAndInvoke()

def failed(self):
    if self.utility.config.Read('failbehavior') == '0':
        # Остановка
        self.utility.actionhandler.procSTOP([self.BitTorrent])
    else:
        # Черга
        self.queueMe()

# Тільки черга, якщо інші закачки чекають

def queueMe(self):
    # дивимосся, коли наступний торрент запускається
    inactiveBitTorrents = self.utility.queue.getInactiveBitTorrents(1)
    if not inactiveBitTorrents:
        return

    nextBitTorrent = inactiveBitTorrents[0]

    # See if this BitTorrent would be started if queued
    queuethis = False

```

```
if (nextBitTorrent.prio < self.BitTorrent.prio):
    queuethis = True
elif (nextBitTorrent.prio == self.BitTorrent.prio) and
(nextBitTorrent.listindex < self.BitTorrent.listindex):
    queuethis = True

if queuethis:
    self.btstatus = self.utility.lang.get('queue')
    self.utility.actionhandler.procQUEUE([self.BitTorrent])
```

K6П3-2023

CH_BitTorrent_client.py - головна програма

```

#!/usr/bin/python

#####
#####
import sys, locale
import os
import wx
#import hotshot

from threading import Thread

from traceback import print_exc
from cStringIO import StringIO

from interconn import ServerListener, ClientPassParam
from launchmanycore import LaunchMany

from CH_BitTorrent_client.Toolbars.toolbars import
CH_BitTorrent_clientBottomBar2, CH_BitTorrent_clientStatusBar,
CH_BitTorrent_clientMenuBar, CH_BitTorrent_clientToolBar
from CH_BitTorrent_client.GUI.menu import CH_BitTorrent_clientMenu
from CH_BitTorrent_client.Scheduler.scheduler import
CH_BitTorrent_clientScheduler

from webservice import WebListener

if (sys.platform == 'win32'):
    from Dialogs.regdialog import RegCheckDialog

from CH_BitTorrent_client.GUI.list import ManagedList
from Utility.utility import Utility
from Utility.constants import * #IGNORE:W0611

#####
#
# Class: FileDropTarget
#
# Щоб вирішити перенесення з подальшим вивільненням для CH_BitTorrent_client
# складають список в головному меню
#
#####
class FileDropTarget(wx.FileDropTarget):
    def __init__(self, utility):
        # ініціалізуємо wsFileDropTarget Object
        wx.FileDropTarget.__init__(self)
        # Запам'ятовуємо значення об'єктів для пропущених файлів
        self.utility = utility

    def OnDropFiles(self, x, y, filenames):
        for filename in filenames:
            self.utility.queue.addBitTorrents.AddBitTorrentFromFile(filename)
        return True

#####
#
# Class : CH_BitTorrent_clientList
#
# CH_BitTorrent_client список класів, які містять список торрентів
#
#####
class CH_BitTorrent_clientList(ManagedList):
    def __init__(self, parent):
        style = wx.LC_REPORT|wx.LC_VRULES|wx.CLIP_CHILDREN

```

```

prefix = 'column'
minid = 4
maxid = 26
exclude = []
rightalign = [COL_PROGRESS,
               COL_SIZE,
               COL_DLSPEED,
               COL_ULSPEED,
               COL_RATIO,
               COL_PEERPROGRESS,
               COL_DLSize,
               COL_ULSize,
               COL_TOTALSPEED]

ManagedList.__init__(self, parent, style, prefix, minid, maxid, exclude,
rightalign)

dragdroplist = FileDropTarget(self.utility)
self.SetDropTarget(dragdroplist)

self.lastcolumnsorted = -1
self.reversesort = False

self.Bind(wx.EVT_KEY_DOWN, self.OnKeyDown)
self.Bind(wx.EVT_LIST_COL_CLICK, self.OnColLeftClick)

self.Bind(wx.EVT_LIST_ITEM_RIGHT_CLICK, self.OnItemSelected)

# Подає додаткові деталі при лівому двійному нажатті миші
self.Bind(wx.EVT_LEFT_DCLICK, self.OnLeftDClick)

# Надає локальні встановлення при натисканні середньої клавіші мишки
self.Bind(wx.EVT_MIDDLE_DCLICK,
self.utility.actions[ACTION_LOCALUPLOAD].action)

# Робить дію, коли натиснута клавіша
def OnKeyDown(self, event):
    keycode = event.GetKeyCode()
    if event.CmdDown():
        if keycode == ord('a') or keycode == ord('A'):
            # виділяє усі файли (CTRL-A)
            self.selectAll()
        elif keycode == ord('x') or keycode == ord('X'):
            # інвертує обрані файли (CTRL-X)
            self.invertSelection()
        elif keycode == wx.WXK_RETURN or keycode == wx.WXK_NUMPAD_ENTER:
            # відкриває додаткові деталі (Enter)
            self.utility.actions[ACTION_DETAILS].action()
        elif keycode == wx.WXK_SPACE:
            # відкриває локальні встановлення (Space)
            self.utility.actions[ACTION_LOCALUPLOAD].action()
        elif keycode == 399:
            # відкриває меню при натисканні правої кнопки миші (windows menu key)
            self.OnItemSelected()

    event.Skip()

def OnColLeftClick(self, event):
    rank = event.GetColumn()
    colid = self.columns.getIDfromRank(rank)
    if colid == self.lastcolumnsorted:
        self.reversesort = not self.reversesort
    else:
        self.reversesort = False
    self.lastcolumnsorted = colid

    self.utility.queue.sortList(colid, self.reversesort)

def selectAll(self):

```

```

self.updateSelected(select = range(0, self.GetItemCount()))

def updateSelected(self, unselect = None, select = None):
    if unselect is not None:
        for index in unselect:
            self.SetItemState(index, 0, wx.LIST_STATE_SELECTED)
    if select is not None:
        for index in select:
            self.Select(index)
    self.SetFocus()

def getBitTorrentSelected(self, firstitemonly = False, reverse = False):
    queue = self.utility.queue

    BitTorrentselected = []
    for index in self.getSelected(firstitemonly, reverse):
        CH_BitTorrent_clientBitTorrentTemp =
queue.getCH_BitTorrent_clientBitTorrent(index = index)
        if CH_BitTorrent_clientBitTorrentTemp is not None:
            BitTorrentselected.append(CH_BitTorrent_clientBitTorrentTemp)
    return BitTorrentselected

def OnItemSelected(self, event = None):
    selected = self.getBitTorrentSelected()
    if not selected:
        return

    popupmenu = CH_BitTorrent_clientMenu(self.utility,
'menu_listrightclick')

    # Випадаюче меню.
    if event is None:
        # використовується позиція першого обраного пункту меню(подія при
натисканні клавіши)
        CH_BitTorrent_clientBitTorrentTemp = selected[0]
        position =
self.GetItemPosition(CH_BitTorrent_clientBitTorrentTemp.listindex)
    else:
        # використовується позиція курсора(подія при натисканні миші)
        position = event.GetPosition()

    self.PopupMenu(popupmenu, position)

def OnLeftDClick(self, event):
    event.Skip()
    self.utility.actions[ACTION_DETAILS].action()

#####
#
# Class : CH_BitTorrent_clientPanel
#
# Головний CH_BitTorrent_client клас панелей інтерфейсу користувача
#
#####
class CH_BitTorrent_clientPanel(wx.Panel):
    def __init__(self, parent):
        style = wx.CLIP_CHILDREN
        wx.Panel.__init__(self, parent, -1, style = style)

        #Відладка вихідних даних.
        sys.stdout.write('Preparing GUI.\n');

        self.utility = parent.utility
        self.utility.window = self
        self.queue = self.utility.queue

        # Список видалених подій торрентів

```

```

self.postponedevents = []

#ручне додавання торрентів UI
#####
colSizer = wx.BoxSizer(wx.VERTICAL)

# Список клавiш управлiння UI
#####
self.list = CH_BitTorrent_clientList(self)
self.utility.list = self.list

colSizer.Add(self.list, 1, wx.EXPAND|wx.ALL, 2)

self.utility.bottomline2 = CH_BitTorrent_clientBottomBar2(self)

colSizer.Add(self.utility.bottomline2, 0, wx.ALL|wx.EXPAND, 3)

self.SetSizer(colSizer)
self.SetAutoLayout(True)

self.list.SetFocus()

def getSelectedList(self, event = None):
    return self.list

#####
# Модифiкацiя CH_BitTorrent_client на льоту
#####
def updateColumns(self, force = False):
    # Модифiкацiя показу у колонках для неактивних торрентiв
    for CH_BitTorrent_clientBitTorrentTemp in
self.utility.BitTorrents["all"]:
        CH_BitTorrent_clientBitTorrentTemp.updateColumns(force = force)

#####
#
# Class : CH_BitTorrent_clientTaskBarIcon
#
# Iконка таскбара
#
#####
class CH_BitTorrent_clientTaskBarIcon(wx.TaskBarIcon):
    def __init__(self, parent):
        wx.TaskBarIcon.__init__(self)

        self.utility = parent.utility

        self.TBMENU_RESTORE = wx.NewId()

        # встановлення iконки таскбара, та вiдстеження подiй вiд нього
        self.Bind(wx.EVT_TASKBAR_LEFT_DCLICK, parent.onTaskBarActivate)
        self.Bind(wx.EVT_MENU, parent.onTaskBarActivate, id =
self.TBMENU_RESTORE)

        self.updateIcon()

def updateIcon(self):
    remove = True

    mintray = self.utility.config.Read('mintray', "int")
    if (mintray >= 2) or ((mintray >= 1) and
self.utility.frame.IsIconized()):
        remove = False

    if remove and self.IsIconInstalled():
        self.RemoveIcon()
    elif not remove and not self.IsIconInstalled():
        self.SetIcon(self.utility.icon, "CH_BitTorrent_client")

```

```

def CreatePopupMenu(self):
    menu = wx.Menu()

    self.utility.actions[ACTION_STOPALL].addToMenu(menu, bindto = self)
    self.utility.actions[ACTION_UNSTOPALL].addToMenu(menu, bindto = self)
    menu.AppendSeparator()
    menu.Append(self.TBMENU_RESTORE,
self.utility.lang.get('showCH_BitTorrent_clientwindow'))
    self.utility.actions[ACTION_EXIT].addToMenu(menu, bindto = self)
    return menu

#####
#
# Class : CH_BitTorrent_clientFrame
#
# Головний CH_BitTorrent_client Фрейм класу, який містить меню й рядок
управління
# та під'єднує CH_BitTorrent_clientPanel
#
#####
class CH_BitTorrent_clientFrame(wx.Frame):
    def __init__(self, ID, params, utility):
        self.utility = utility
        self.utility.frame = self

        title = self.utility.lang.get('title') + \
            " " + \
            self.utility.lang.get('version')

        # Беремо розмір вікна та позиціонуємо у конфігураційному файлі
        size, position = self.getWindowSettings()
        style = wx.DEFAULT_FRAME_STYLE | wx.CLIP_CHILDREN
        wx.Frame.__init__(self, None, ID, title, position, size, style = style)

        self.tbicon = None

        self.CH_BitTorrent_client_sb = CH_BitTorrent_clientStatusBar(self)
        self.SetStatusBar(self.CH_BitTorrent_client_sb)

        try:
            self.SetIcon(self.utility.icon)
        except:
            pass

        # не модифікуєм GUI
        self.GUIмодифікація = True

        # Запускаємо планувальник перед створенням ListCtrl
        self.utility.queue = CH_BitTorrent_clientScheduler(self.utility)

        self.window = CH_BitTorrent_clientPanel(self)

        # Меню опцій
        #####
        menuBar = CH_BitTorrent_clientMenuBar(self)
        self.SetMenuBar(menuBar)

        self.tb = CH_BitTorrent_clientToolBar(self)
        self.SetToolBar(self.tb)

        # Меню подій
        #####

        self.Bind(wx.EVT_CLOSE, self.OnCloseWindow)
#         self.Bind(wx.EVT_MENU, self.OnMenuExit, id = wx.ID_CLOSE)

```

```

self.Bind(wx.EVT_QUERY_END_SESSION, self.OnCloseWindow)
self.Bind(wx.EVT_END_SESSION, self.OnCloseWindow)

try:
    self.tbicon = CH_BitTorrent_clientTaskBarIcon(self)
except:
    pass
self.Bind(wx.EVT_ICONIZE, self.onIconify)
self.Bind(wx.EVT_SET_FOCUS, self.onFocus)

# перевіряємо вебсервіси для автостарту вебсервісів
#####
WebListener(self.utility)
if self.utility.webconfig.Read("webautostart", "boolean"):
    self.utility.webserver.start()

# Запускаємо диспетчера
self.utility.controller = LaunchMany(self.utility)
self.utility.controller.start()

self.utility.queue.postInitTasks()

# Запускаємо єдиний сервер listener
#####
self.serverlistener = ServerListener(self.utility)
self.serverthread = Thread(target = self.serverlistener.start)
self.serverthread.setDaemon(False)
self.serverthread.start()

#Якщо сервер запускається з заданими параметрами
#####
if params[0] != "":
    ClientPassParam(params[0])

sys.stdout.write('GUI Complete.\n')

self.Show(True)

# перевіряємо, якщо CH_BitTorrent_client підключився до торрентів
#####
if (sys.platform == 'win32'):
    if self.utility.config.Read('associate', "boolean"):
        if not self.utility.regchecker.testRegistry():
            dialog = RegCheckDialog(self)
            dialog.ShowModal()
            dialog.Destroy()

def onFocus(self, event = None):
    if event is not None:
        event.Skip()
    self.window.getSelectedList(event).SetFocus()

def setGUIupdate(self, update):
    oldval = self.GUIupdate
    self.GUIмодифікація = update

    if self.GUIмодифікація and not oldval:
        # Force an модифікація of all BitTorrents
        for BitTorrent in self.utility.BitTorrents["all"]:
            BitTorrent.updateColumns()
            BitTorrent.updateColor()

#####
# мінімізуємо до трьох панелей управління
#####
def onTaskBarActivate(self, event = None):
    self.Iconize(False)
    self.Show(True)

```

```

self.Raise()

if self.tbicon is not None:
    self.tbicon.updateIcon()

self.window.list.SetFocus()

# Підсумкове оновлення GUI
self.setGUIupdate(True)

def onIconify(self, event = None):
    if (self.utility.config.Read('mintray', "int") > 0
        and self.tbicon is not None):
        self.tbicon.updateIcon()
        self.Show(False)

    if event is not None:
        event.Skip()

    # Не модифікуємо GUI поки він мінімізований
    self.setGUIupdate(not self.GUIupdate)

def getWindowSettings(self):
    width = self.utility.config.Read("window_width")
    height = self.utility.config.Read("window_height")
    try:
        size = wx.Size(int(width), int(height))
    except:
        size = wx.Size(710, 400)

    x = self.utility.config.Read("window_x")
    y = self.utility.config.Read("window_y")
    if (x == "" or y == ""):
        position = wx.DefaultPosition
    else:
        position = wx.Point(int(x), int(y))

    return size, position

def saveWindowSettings(self):
    width, height = self.GetSizeTuple()
    x, y = self.GetPositionTuple()
    self.utility.config.Write("window_width", width)
    self.utility.config.Write("window_height", height)
    self.utility.config.Write("window_x", x)
    self.utility.config.Write("window_y", y)

    self.utility.config.Flush()

#####
# Закриття програми
#####

def OnCloseWindow(self, event = None):
    # Нічого не робимо, якщо подія викликана двічі
    if self.utility.CH_BitTorrent_clientquitting:
        return

    # Перевіряємо, щоб поглянути якщо ми забороняємо перезавантаження
    try:
        if event.CanVeto() and
self.utility.config.Read('confirmonclose', "boolean"):
            dialog = wx.MessageDialog(None,
self.utility.lang.get('confirmmsg'), self.utility.lang.get('confirm'),
wx.OK|wx.CANCEL)

            result = dialog.ShowModal()
            dialog.Destroy()
            if result != wx.ID_OK:
                event.Veto()

```

```

        return

    except:
        data = StringIO()
        print_exc(file = data)
        sys.stderr.write(data.getvalue())
        pass

self.utility.CH_BitTorrent_clientquitting = True
self.GUImодифікація = False

# Закриваємо створений торрент
self.utility.actions[ACTION_MAKEBITTORRENT].closeWin()

try:
    self.utility.webserver.stop()
except:
    data = StringIO()
    print_exc(file = data)
    sys.stderr.write(data.getvalue())
    pass

try:
    # встановлюємо планувальник на закриття усіх активних торрентів
    self.utility.queue.clearScheduler()
except:
    data = StringIO()
    print_exc(file = data)
    sys.stderr.write(data.getvalue())
    pass

try:
    # Зберігаємо вікно після збереження розміру та позиції
    self.onTaskBarActivate()
    self.saveWindowSettings()
except:
    data = StringIO()
    print_exc(file = data)
    sys.stderr.write(data.getvalue())
    pass

try:
    if self.tbicon is not None:
        self.tbicon.RemoveIcon()
        self.tbicon.Destroy()
    self.Destroy()
except:
    data = StringIO()
    print_exc(file = data)
    sys.stderr.write(data.getvalue())
    pass

#####
#
# Class : CH_BitTorrent_clientApp
#
# Головний CH_BitTorrent_client клас додатку під'єднуючий
CH_BitTorrent_clientFrame Object
#
#####
class CH_BitTorrent_clientApp(wx.App):
    def __init__(self, x, params, single_instance_checker,
CH_BitTorrent_clientpath):
        self.params = params
        self.single_instance_checker = single_instance_checker

self.utility = Utility(CH_BitTorrent_clientpath)
# встановлюємо куди завантажувати дані з торренту
locale.setlocale(locale.LC_ALL, '')

```

```

sys.stdout.write('Client Starting Up.\n')
sys.stdout.write('Build: ' + self.utility.lang.get('build') + '\n')

wx.App.__init__(self, x)

def OnInit(self):
    self.utility.postAppInit()
    self.frame = CH_BitTorrent_clientFrame(-1, self.params, self.utility)

    self.Bind(wx.EVT_QUERY_END_SESSION, self.frame.OnCloseWindow)
    self.Bind(wx.EVT_END_SESSION, self.frame.OnCloseWindow)

    return True

def OnExit(self):
    del self.single_instance_checker
    ClientPassParam("Закриття з'єднання")

    return 0

#####
#
# Тут починається головна програма
#
#####
def run(params = None):
    if params is None:
        params = [""]

    if len(sys.argv) > 1:
        params = sys.argv[1:]

    # Створюємо єдиний маркер для операцій
    single_instance_checker = wx.SingleInstanceChecker("pingpong-
CH_BitTorrent_client" + wx.GetUserId())

    if single_instance_checker.IsAnotherRunning():
        #Відправляємо інформацію про торрент до CH_BitTorrent_client
        ClientPassParam(params[0])
    else:
        CH_BitTorrent_clientpath = os.path.abspath(os.path.dirname(sys.argv[0]))
        os.chdir(CH_BitTorrent_clientpath)

        # Почнемо спочатку CH_BitTorrent_client єдиного зразка
        app = CH_BitTorrent_clientApp(0, params, single_instance_checker,
CH_BitTorrent_clientpath)
        app.MainLoop()

if __name__ == '__main__':
    run()

```

interconn.py - З'єднання з торрент-трекером

```

import wx
import sys

from Utility.helpers import getSocket

#####
#
# Class: ServerListener
#
# Єдиного зразка відкритий локальний порт
#
#####
class ServerListener:
    def __init__(self, utility):
        self.s = None
        self.utility = utility

    def start(self):
        HOST = '127.0.0.1'          # символічне ім'я локального хосту

        PORT = 56666              # Визначає який порт більш привілейований
        self.s = getSocket(HOST, PORT, "server")
        if self.s is None:
            sys.stderr.write('Не можу відкрити сокет') # No way
            sys.exit(1)
        while 1:
            try:
                conn, addr = self.s.accept()
                data = conn.recv(1024)
                conn.close()
                # якщо прапор черги встановлений, ми робимо запит на закриття
                if self.utility.CH_BitTorrent_clientquitting or data ==
"Закриття з'єднання":
                    self.s.close()
                    break
            # elif data == "KEEPALIVE":
            #     # переконавшись що з'єднання за час виділений для цього не
            #     встановилось...
            #     pass
            elif data == "Наступне вікно":
                self.utility.frame.onTaskBarActivate()
            else:
                self.utility.queue.addBitTorrents.AddBitTorrentFromFile(data)
                except wx.PyDeadObjectError:
                    toosooncontext = "\n Пробоємо стартувати CH_BitTorrent_client знову
після завершення! \n" + \
                                "(Чекаємо CH_BitTorrent_client поки він закінчить
працювати)\n"
                    sys.stderr.write(toosooncontext)
                    break
            #     self.utility.CH_BitTorrent_clientdonequitting = True
            #     sys.stderr.write("\n Не перезавантажується список серверів ")

#####
#
# Class: ClientPassParam
#
# Інші торренти окрім сервера передають параметр
#
#####
def ClientPassParam(params, ignoreError = False):
    HOST = '127.0.0.1'          # Видаляємо хост

```

```
PORT = 56666 # Такий же порт, як й той, що
використовується сервером
# Повторюємо спробу ...
s = getSocket(HOST, PORT, 15)
if s is None:
    if ignoreError:
        return
    print 'не можу відкрити сокет'
#     sys.stderr.write("Не можу відкрити сокет\n")
#     sys.stderr.write("--Спроба переслати параметри: (" + params + ")\n")
    sys.exit(1)

# Якщо не закінчилося з'єднання
#####
if not params:
    s.send("Переходимо до головного вікна")
else:
    s.send(params)
s.close()
```

КБГПЗ - 2023

webservice.py - веб-сервіси

```

import wx
import sys

from threading import Thread
from string import upper

from traceback import print_exc
from cStringIO import StringIO

from Utility.helpers import getSocket
from Utility.constants import * #IGNORE:W0611

#####
#
# Class: WebDialog
#
# Встановлення конкретних установок користувача для веб-сервісу
#
#####
class WebDialog(wx.Dialog):
    def __init__(self, parent):
        self.utility = parent.utility

        title = self.utility.lang.get('webinterfacesservice')

        pre = wx.PreDialog()
        pre.Create(parent, -1, title)
        self.this = pre.this
        self.parent = parent
        self.window = self.utility.window

        self.utility.webserver.webdlg = self

        WebRead = self.utility.webconfig.Read

        self.warnlowport = [False, WebRead('webport')]

        # Змінюємо старі конфігураційні значення
        # oldval = WebRead('webIP')
        # if (oldval == "Automatic") or (oldval == "Automatic"):
        #     self.utility.webconfig.Write('webIP', "")
        #     self.utility.webconfig.Flush()
        #     ip_choice = ""
        # elif (oldval == "Loop Back"):
        #     self.utility.webconfig.Write('webIP', "127.0.0.1")
        #     self.utility.webconfig.Flush()
        #
        #     newval = WebRead('webIP')
        newval = self.utility.webserver.getIP()
        if newval == "":
            default_ip = self.utility.lang.get('automatic')
        elif newval == "127.0.0.1":
            default_ip = self.utility.lang.get('loopback')
        else:
            default_ip = newval

        outerbox = wx.BoxSizer(wx.VERTICAL)

        outerbox.Add(wx.StaticText(self, -1,
self.utility.lang.get('webinterfacetitle')), 0, wx.ALIGN_CENTER|wx.ALL, 10)

        ip_choices = [self.utility.lang.get('automatic'),
self.utility.lang.get('loopback')]

```

```

        ipandport_box = wx.BoxSizer(wx.HORIZONTAL)
        ipandport_box.Add(wx.StaticText(self, -1,
self.utility.lang.get('webip')), 0, wx.ALIGN_CENTER_VERTICAL|wx.ALL, 5)
        self.ipertext = wx.ComboBox(self, -1, default_ip, wx.Point(-1, -1),
wx.Size(100, -1), ip_choices, wx.CB_DROPDOWN|wx.CB_READONLY)
        ipandport_box.Add(self.ipertext, 0, wx.ALIGN_CENTER_VERTICAL|wx.ALL, 5)

        ipandport_box.Add(wx.StaticText(self, -1,
self.utility.lang.get('webport')), 0, wx.ALIGN_CENTER_VERTICAL|wx.ALL, 5)
        self.porttext = self.utility.makeNumCtrl(self, WebRead('webport'), max =
65536)
        ipandport_box.Add(self.porttext, 0, wx.ALIGN_CENTER_VERTICAL|wx.ALL, 5)

        outerbox.Add(ipandport_box, 0, wx.EXPAND|wx.ALL, 5)

        uniquekey_box = wx.BoxSizer(wx.HORIZONTAL)

        uniquekey_box.Add(wx.StaticText(self, -1,
self.utility.lang.get('uniquekey')), 0, wx.ALIGN_CENTER_VERTICAL|wx.ALL, 5)
        self.keytext = wx.TextCtrl(self, -1, WebRead('webID'), wx.Point(-1, -1),
wx.Size(165, -1))
        uniquekey_box.Add(self.keytext, 0, wx.ALIGN_CENTER_VERTICAL|wx.ALL, 5)

        outerbox.Add(uniquekey_box, 0, wx.EXPAND|wx.LEFT|wx.RIGHT, 5)

        self.permissions = [ ['webquery', 'allow_query'],
                             ['webadd', 'allow_add'],
                             ['webdelete', 'allow_delete'],
                             ['webqueue', 'allow_queue'],
                             ['webstop', 'allow_stop'],
                             ['webresume', 'allow_resume'],
                             ['webpause', 'allow_pause'],
                             ['webclearallcompleted', 'allow_clearcompleted'],
                             ['priority', 'allow_setprio'],
                             ['webgetparam', 'allow_getparam'],
                             ['websetparam', 'allow_setparam'] ]

        self.perm_checkbox = wx.CheckListBox(self, -1, size = wx.Size(100, 120),
style = wx.LB_SINGLE)
        perm_text = [self.utility.lang.get(item[0]) for item in
self.permissions]
        self.perm_checkbox.Set(perm_text)

        for i in range(0, len(self.permissions)):
            param = self.permissions[i][1]
            checked = WebRead(param, "boolean")
            if checked:
                self.perm_checkbox.Check(i)

        outerbox.Add(wx.StaticText(self, -1,
self.utility.lang.get('commandpermission')), 0, wx.ALL, 5)
        outerbox.Add(self.perm_checkbox, 0, wx.EXPAND|wx.LEFT|wx.RIGHT, 5)

        self.webautostart = wx.CheckBox(self, -1,
self.utility.lang.get('webautostart'))

        outerbox.Add(self.webautostart, 0, wx.EXPAND|wx.ALL, 5)

        #CheckBox Встановлює значення з конфігураційного файлу
        #####
        self.webautostart.SetValue(WebRead('webautostart', "boolean"))

        self.actionbtn = wx.Button(self, -1, "")

        self.utility.webserver.updateLabels()

        applybtn = wx.Button(self, wx.NewId(), self.utility.lang.get('apply'))
        self.Bind(wx.EVT_BUTTON, self.onApply, applybtn)

```

```

okbtn = wx.Button(self, wx.NewId(), self.utility.lang.get('ok'))
self.Bind(wx.EVT_BUTTON, self.onOK, okbtn)

cancelbtn = wx.Button(self, wx.ID_CANCEL,
self.utility.lang.get('cancel'))
self.Bind(wx.EVT_BUTTON, self.onClose, cancelbtn)

self.Bind(wx.EVT_CLOSE, self.onClose)

self.Bind(wx.EVT_BUTTON, self.OnAction, self.actionbtn)

button_box = wx.BoxSizer(wx.HORIZONTAL)
button_box.Add(applybtn, 0, wx.ALL, 5)
button_box.Add(okbtn, 0, wx.ALL, 5)
button_box.Add(cancelbtn, 0, wx.ALL, 5)

outerbox.Add(self.actionbtn, 0, wx.ALIGN_CENTER|wx.ALL, 5)

outerbox.Add(button_box, 0, wx.ALIGN_CENTER)

self.SetAutoLayout(True)
self.SetSizer(outerbox)
self.Fit()

def onOK(self, event = None):
self.utility.webserver.webdlg = None
if self.onApply():
self.EndModal(wx.ID_OK)

def onClose(self, event = None):
self.utility.webserver.webdlg = None
if event is not None:
event.Skip()

def onApply(self, event = None):
self.saveParams()
return True

def saveParams(self):
PORT = int(self.porttext.GetValue())

if ((not self.warnlowport[0]
or self.warnlowport[1] != str(PORT))
and PORT < 1024):
dlg = wx.MessageDialog(None,
self.utility.lang.get('warningportunder1024') + "\n(" + str(PORT) + ")",
self.utility.lang.get('warning'), wx.YES_NO|wx.ICON_INFORMATION)
if dlg.ShowModal() == wx.ID_NO:
dlg.Destroy()
return False
else:
self.warnlowport[0] = True
self.warnlowport[1] = str(PORT)
dlg.Destroy()

# Повторно призначаемо веб параметри
self.utility.webconfig.Write('webIP', self.iptext.GetStringSelection())
self.utility.webconfig.Write('webport', str(self.porttext.GetValue()))
self.utility.webconfig.Write('webID', self.keytext.GetValue())

for i in range(0, self.perm_checkbox.GetCount()):
param = self.permissions[i][1]
checked = self.perm_checkbox.IsChecked(i)
self.utility.webconfig.Write(param, checked, "boolean")

if self.webautostart.GetValue():
self.utility.webconfig.Write('webautostart', "1")
else:
self.utility.webconfig.Write('webautostart', "0")

```

```

#####
# Записуємо нову конфігурацію до webservice.conf
#####
self.utility.webconfig.Flush()

return True

def OnAction(self, event = None):
    if not self.utility.webserver.active:
        if self.saveParams():
            self.startWebService()
    else:
        self.stopWebService()

def startWebService(self):
    #####
    # Зануємо Web Interface Service
    #####
    self.utility.webserver.start()

def stopWebService(self):
    # Зупиняємо Web Service
    self.utility.webserver.stop()

def updateLabels(self):
    active = self.utility.webserver.active

    if active:
        label = self.utility.lang.get('stopservice')
    else:
        label = self.utility.lang.get('startservice')
    self.actionbtn.SetLabel(label)
    self.porttext.Enable(not active)
    self.keytext.Enable(not active)
    self.iptext.Enable(not active)

#####
#
# Class: WebListener
#
# Список команд веб сервісу
#
#####
class WebListener:
    def __init__(self, utility, webdlg = None):
        self.s = None
        self.utility = utility

        self.utility.webserver = self

        self.webdlg = webdlg

        self.active = False

        self.port = None
        self.ip = None

        self.client = WebClient(self.utility)

def getIP(self):
    # Змінюємо старі конфігураційні значення
    oldval = self.utility.webconfig.Read('webIP')
    if (oldval == "Automatics") or (oldval == "Automatic"):
        self.utility.webconfig.Write('webIP', "")
        self.utility.webconfig.Flush()
    elif (oldval == "Loop Back"):
        self.utility.webconfig.Write('webIP', "127.0.0.1")

```

```

        self.utility.webconfig.Flush()
#       if self.utility.webconfig.Read('webIP') == "Automatic":
#           IP = ""
#       else: #LoopBack
#           IP = "127.0.0.1"
#       return IP
return self.utility.webconfig.Read('webIP')

def getPort(self):
    try:
        PORT = self.utility.webconfig.Read('webport', "int")
    except:
        PORT = 56667
    return PORT

def start(self):
    # Already running
    if self.active:
        return

    self.ip = self.getIP() # Символьне ім'я локального хосту
    self.port = self.getPort() # Визначаємо який порт має більший привілей

    self.active = True
    self.updateLabels()

    webservice = Thread(target = self.startThread)
    webservice.setDaemon(False)
    webservice.start()

def stop(self):
    # Не запускаємо, немає необхідності у зупинці
    if not self.active:
        return

    self.client.sendCmd("Закриваємо|")

    self.port = None
    self.ip = None

    self.active = False
    self.updateLabels()

def updateLabels(self):
    self.utility.actions[ACTION_WEBSERVICE].updateButton()

    if self.webdlg is not None:
        self.webdlg.updateLabels()

def startThread(self):
    self.s = getSocket(self.ip, self.port, "server")

    if self.s is None:
        #Виводимо повідомлення, що сокет не відкрито
        dlg = wx.MessageDialog(None, self.utility.lang.get('cantopensocket')
, self.utility.lang.get('socketerror'), wx.OK|wx.ICON_ERROR)
        dlg.ShowModal()
        dlg.Destroy()

        self.active = False
        self.updateLabels()
        return
    WebServiceCmd(self).go()

#####
#
# Class: WebClient
#

```

```

# Використовуємо посилання коротких команд до веб сервіса
# (використовуємо посилання команд перезавантаження)
#
#####
class WebClient:
    def __init__(self, utility):
        self.utility = utility
        self.webserver = self.utility.webserver

    def sendCmd(self, command):
        # Веб сервіс не активний - надсилаємо відповідне повідомлення
        if not self.webserver.active:
            return False

        HOST = self.webserver.ip           # Видаляємо хост
        PORT = self.webserver.port         # Такий же порт, як й той, що
        використовується сервером
        s = getSocket(HOST, PORT)
        if s is None:
            dlg = wx.MessageDialog(None,
self.utility.lang.get('cantconnectCH_BitTorrent_clientwebinterface') ,
self.utility.lang.get('socketerror'), wx.OK|wx.ICON_ERROR)
            dlg.ShowModal()
            dlg.Destroy()
            return False

        # Якщо запит не закрив запит з'єднання
        # то цей запит тор рента копіюємо до .BitTorrent
        # в папці торрента
        #####
        mesg = "ID|" + self.utility.webconfig.Read('webID') + "\n" + command

        s.send(mesg)
        s.close()
        return True

#####
#
# Class: WebServiceCmd
#
# Обробка фактичних команд веб сервіса
#
#####
class WebServiceCmd:
    def __init__(self, parent):
        self.parent = parent
        self.utility = self.parent.utility
        self.frame = self.utility.frame

    def separate(self, info, splitchar = ","):
        try:
            separated = info.split(splitchar)
        except:
            self.conn.send("Повертає \n помилку = Погані аргументи")
            self.conn.close()
            separated = None
        return separated

    def getMappings(self, separated, splitchar = " = "):
        mappings = {}

        for item in separated:
            try:
                pair = item.split(splitchar)
                mappings[pair[0]] = pair[1]
            except:
                self.conn.send("Повертає \n помилку = Погані аргументи")
                self.conn.close()

```

```

        mappings = None
        break

    return mappings

def getBitTorrents(self, info = "", infohash_list = None):
    BitTorrents = None

    if infohash_list is None:
        infohash_list = self.separate(info)
        if infohash_list is None:
            return BitTorrents

    BitTorrents = []

    for infohash in infohash_list:
        BitTorrent =
self.utility.queue.getCH_BitTorrent_clientBitTorrent(info_hash = infohash)

        if BitTorrent is None:
            self.conn.send("Повертає \n помилку = Немає торренту з цією хеш
інформацією: " + infohash + "\n")
            BitTorrents = None
            break
        else:
            BitTorrents.append(BitTorrent)

    return BitTorrents

def cmdSetParam(self, info):
    conn = self.conn

    if not self.utility.webconfig.Read('allow_setparam', "boolean"):
        conn.send("Повертає \n помилку = SETPARAM,Відхилений дозвіл")
        return

    separated = self.separate(info, "|")
    if separated is None:
        return

    mappings = self.getMappings(separated)
    if mappings is None:
        return

    for param in mappings:
        self.utility.config.Write(param, mappings[param])
    self.utility.config.Flush()
    conn.send("Повертає \n ОК")

def cmdGetParam(self, info):
    conn = self.conn

    if not self.utility.webconfig.Read('allow_getparam', "boolean"):
        conn.send("Повертає \n помилку = GETPARAM,Відхилений дозвіл")
        return

    separated = self.separate(info)
    if separated is None:
        return

    retmsg = ""
    Read = self.utility.config.Read
    for param in separated:
        value = Read(param)
        retmsg += value + "|"
    retmsg += "\n"
    conn.send("Повертає \n " + str(retmsg))

def cmdClose(self):

```

```

conn = self.conn

self.utility.webserver.active = False
self.utility.webserver.updateLabels()

conn.close()
self.parent.s.close()

# sys.stderr.write("\nНе може перезавантажити веб сервіс")

def cmdQuery(self, info = ""):
    conn = self.conn

    if not self.utility.webconfig.Read('allow_query', "boolean"):
        conn.send("Повертає \n помилку = QUERY, Відхилений дозвіл")
        conn.close()
        return

    if info == "":
        self.QueryFields()
    else:
        fields = self.separate(info, ",")
        if fields is None:
            return

        self.QueryFields(fields)

def QueryFields(self, fieldlist = None):
    conn = self.conn

    maxid = self.utility.list.columns.maxid

    # Типове значення до повернення всіх полів
    if fieldlist is None:
        fieldlist = range(4, maxid)

    oldfields = { "filename"      : COL_TITLE,
                  "progress"     : COL_PROGRESS,
                  "btstatus"     : COL_BTSTATUS,
                  "eta"          : COL_ETA,
                  "dlspeed"      : COL_DLSPEED,
                  "ulspeed"      : COL_ULSPEED,
                  "ratio"        : COL_RATIO,
                  "peers"        : COL_PEERS,
                  "seeds"        : COL_SEEDS,
                  "copies"       : COL_COPIES,
                  "dlspeed"      : COL_DLSPEED,
                  "ulsize"       : COL_ULSIZE,
                  "peeravgprogress" : COL_PEERPROGRESS,
                  "totalspeed"   : COL_TOTALSPEED,
                  "totalsize"    : COL_SIZE,
                  "priority"    : COL_PRIO }

    fieldids = []

    retmsg = ""
    for req_field in fieldlist:
        try:
            # Номер поля був вказаний
            fieldid = int(req_field)
            if fieldid >= 4 and fieldid < maxid:
                fieldids.append(fieldid)
            else:
                conn.send("Повертає \n помилку = Помилкове поле ID (повино
бути між 4 та " + str(maxid) + ") = " + req_field)
                conn.close()
                return
        except:
            # Старий формат-- ім'я поля було вказане

```

```

        if req_field in oldfields:
            fieldid = oldfields[req_field]
            fieldids.append(fieldid)
        else:
            # Can't identify the field
            conn.send("Повертає \n помилку = Невідоме поле імені = " +
req_field)

            conn.close()
            return

        retmsg += self.utility.lang.get("column" + str(fieldid) + "_text")
+ "|"

        retmsg += "Info Hash\n"

        for CH_BitTorrent_clientBitTorrentTemp in
self.utility.BitTorrents["all"]:
            retmsg += CH_BitTorrent_clientBitTorrentTemp.getInfo(fieldlist =
fieldids)

            conn.send(retmsg)
            conn.close()

    def cmdAdd(self, info):
        conn = self.conn

        if not self.utility.webconfig.Read('allow_add', "boolean"):
            conn.send("Повертає \n помилку = ADD,Відхилений дозвіл")
            return

        # Що ми робимо, якщо ми не маємо розташування довільного завантаження,
        конкретизували, і ми звертаємося до цього з веб сервіса
        #####
        retmsg = self.utility.queue.addBitTorrents.AddBitTorrentURL(info, "web")
        conn.send("Повертає \n "+retmsg)

    def cmdDelete(self, info):
        conn = self.conn

        if upper(info) == "COMPLETED":
            if not self.utility.webconfig.Read('allow_clearcompleted',
"boolean"):
                conn.send("Повертає \n помилку = CLEARCOMPLETED,Відхилений
дозвіл")
                return

            self.utility.actions[ACTION_CLEARCOMPLETED].action()
            conn.send("Повертає \n ОК")
        else:
            if not self.utility.webconfig.Read('allow_delete', "boolean") !=
"1":
                conn.send("Повертає \n помилку = DELETE,Відхилений дозвіл")
                return

            BitTorrents = self.getBitTorrents(info)
            if BitTorrents is None:
                return

            self.utility.actionhandler.procREMOVE(BitTorrents)
            conn.send("Повертає \n ОК")

    def cmdResume(self, info = "ALL"):
        conn = self.conn

        if not self.utility.webconfig.Read('allow_resume', "boolean"):
            conn.send("Повертає \n помилку = RESUME,Відхилений дозвіл")
            return

        if upper(info) == "ALL":

```

```

        self.utility.actions[ACTION_UNSTOPALL].action()
        conn.send("Повертає \n OK")
    else:
        BitTorrents = self.getBitTorrents(info)
        if BitTorrents is None:
            return

        self.utility.actionhandler.procRESUME(BitTorrents)
        conn.send("Повертає \n OK")

def cmdStop(self, info = "ALL"):
    conn = self.conn

    if not self.utility.webconfig.Read('allow_stop', "boolean"):
        conn.send("Повертає \n помилку = Зупинка, Відхилений дозвіл")
        return

    if upper(info) == "ALL":
        self.utility.actions[ACTION_STOPALL].action()
        conn.send("Повертає \n OK")
    else:
        BitTorrents = self.getBitTorrents(info)
        if BitTorrents is None:
            return

        self.utility.actionhandler.procSTOP(BitTorrents)
        conn.send("Повертає \n OK")

def cmdPause(self, info, release = False):
    conn = self.conn

    if not self.utility.webconfig.Read('allow_pause', "boolean"):
        conn.send("Повертає \n помилку = Пауза , Відхилений дозвіл")
        conn.close()
        return

    if upper(info) == "ALL":
        BitTorrents = self.utility.BitTorrents["active"].keys()
    else:
        BitTorrents = self.getBitTorrents(info)
        if BitTorrents is None:
            return

    self.utility.actionhandler.procPAUSE(BitTorrents, release = release)
    conn.send("Повертає \n OK")

def cmdQueue(self, info = "ALL"):
    conn = self.conn

    if not self.utility.webconfig.Read('allow_queue', "boolean"):
        conn.send("Повертає \n помилку = Черга, Відхилений дозвіл")
        return

    if upper(info) == "ALL":
        BitTorrents = self.utility.BitTorrents["all"]
    else:
        BitTorrents = self.getBitTorrents(info)
        if BitTorrents is None:
            return

    self.utility.actionhandler.procQUEUE(BitTorrents)
    conn.send("Повертає \n OK")

def cmdPriority(self, info):
    conn = self.conn

    if not self.utility.webconfig.Read('allow_setprio', "boolean"):
        conn.send("Повертає \n помилку = Пріоритет, Відхилений дозвіл")
        return

```

```

separated = self.separate(info, "|")
if separated is None:
    return

mappings = self.getMappings(separated, ",")
if mappings is None:
    return

for infohash in mappings:
    value = mappings[infohash]
    error = False
    try:
        prio = int(value)
        if (prio < 0) or (prio > 4):
            error = True
    except:
        error = True
    if error:
        conn.send("Повертає \n помилку = Пріоритет має бути номером 0-4")

        conn.close()
        return

hashlist = [infohash for infohash in mappings]
BitTorrents = self.getBitTorrents(infohash_list = hashlist)
if BitTorrents is None:
    return

for BitTorrent in BitTorrents:
    BitTorrent.changePriority(prio)

conn.send("Повертає \n OK")

def go(self):
    while 1:
        conn, addr = self.parent.s.accept()
        self.conn = conn
        try:
            try:
                data = conn.recv(5048)
            except:
                try:
                    conn.close()
                except:
                    pass
                continue

            try:
                idline, cmdline = data.split("\n")
            except:
                conn.send("Повертає \n помилку = Вам потрібні унікальні ID до управління CH_BitTorrent_client!")
                conn.close()
                continue

            try:
                idtag = idline.split("|")
                idkey = idtag[0]
                idvalue = idtag[1]
            except:
                conn.send("Повертає \n помилку = Вам потрібні унікальні ID до управління CH_BitTorrent_client!")
                conn.close()
                continue

            if idkey != "ID":
                conn.send("Повертає \n помилку = Вам потрібні унікальні ID до управління CH_BitTorrent_client!")

```

```

        conn.close()
        continue
    else:
        if idvalue != self.utility.webconfig.Read('webID'):
            conn.send("Повертає \n помилку = Некоректний унікальний
ID")

            conn.close()
            continue

    try:
        cmd, info = cmdline.split("|")
    except:
        #Погана команда
        conn.send("Повертає \n помилку = Команда повинна закінчитися
|")

        conn.close()
        continue

    sys.stdout.write(' Веб запит переданий .\n' + data + '\n')

    # Перетворюємо команду на верхній регістр
    cmd = upper(cmd)

    if cmd == "CLOSE" or
self.utility.CH_BitTorrent_clientquitting:
        self.cmdClose()
        return

#
#
        elif cmd == "KEEPALIVE":
            pass

        elif cmd == "QUERY":
            self.cmdQuery(info)

        elif cmd == "ADD":
            self.cmdAdd(info)

        elif cmd == "DELETE":
            self.cmdDelete(info)

        elif cmd == "RESUME":
            self.cmdResume(info)

        elif cmd == "PAUSE":
            self.cmdPause(info)

        elif cmd == "UNPAUSE":
            self.cmdPause(info, release = True)

        elif cmd == "STOP":
            self.cmdStop(info)

        elif cmd == "QUEUE":
            self.cmdQueue(info)

        elif cmd == "PRIORITY":
            self.cmdPriority(info)

        elif cmd == "SETPARAM":
            self.cmdSetParam(info)

        elif cmd == "GETPARAM":
            self.cmdGetParam(info)

        elif cmd == "GETSTRING":
            conn.send("String\n" + self.utility.lang.get(info))

        elif cmd == "VERSION":
            conn.send("Version\n" + self.utility.lang.get('version'))

```

```
else: # Погана команда
    conn.send("Повертає \n помилку = Команда не знайдена: " +
cmd)
        pass
except:
    pass
conn.close()
```

КБПЗ - 2023

webtest.py - тестування програми

```

from sys import argv, exit
from time import sleep

from Utility.helpers import getSocket

#####
#
# Class: WebClient
#
# Використовує відправку команд до веб сервісу
#
#####
class WebClient:
    def __init__(self, ip, port):
        self.ip = ip
        self.port = port

    def sendmesg(self, mesg):
        HOST = self.ip          # Видаляємо хост
        PORT = self.port        # Такий же порт, як й той, що використовується
сервером
        s = getSocket(HOST, PORT)
        if s is None:
            return "Не з'єднано CH_BitTorrent_client з веб сервісом"

        # якщо запит - не запит з'єднання
        # тоді цей запит торренту записуємо до .BitTorrent
        # у папці торрента
        #####
        s.send(mesg)
        retmesg = s.recv(5000)
        s.close()
        return retmesg

##### Функція тестування #####
if len(argv) != 4:
    print "Використовуємо: webtest.py <unique key> <IP> <Port>"
    exit(0)

KEY      = argv[1]
IP       = argv[2]
PORT     = argv[3]

wc = WebClient(argv[2], argv[3])
print "----- Починаємо тестування веб сервісу-----"
print "Черга команд: "
ret = wc.sendmesg("ID|" + KEY + "\nQUERY|")
line = ret.split("\n")
key = line[0].split("|")
dict = {}
for i in key:
    dict[i] = ""
if len(line) > 1:
    for i in range(1, len(line)-1):
        result = line[i].split("|")
        for j in range(0, len(key)):
            dict[key[j]] = result[j]
            print key[j] + " = " + dict[key[j]]
    print "-----"

print "-----"
sleep(5)
print "Додаємо команду: "

```

```

print wc.sendmsg("ID|" + KEY + "\nADD|BitTorrenturlthatworks")
print "-----"
sleep(5)

print "Зупинка команди: "
print wc.sendmsg("ID|" + KEY +
"\nSTOP|a5051a665c837d56b21f5d612e15e9992fe68f27")
print "-----"
sleep(5)

print "Черга команд: "
print wc.sendmsg("ID|" + KEY +
"\nQUEUE|a5051a665c837d56b21f5d612e15e9992fe68f27")
print "-----"
sleep(5)

print "Зупинка команди: "
print wc.sendmsg("ID|" + KEY +
"\nSTOP|a5051a665c837d56b21f5d612e15e9992fe68f27")
print "-----"
sleep(5)

print "Підсумкова команда: "
print wc.sendmsg("ID|" + KEY +
"\nRESUME|a5051a665c837d56b21f5d612e15e9992fe68f27")
print "-----"
sleep(5)

print "Добре завершена команда: "
print wc.sendmsg("ID|" + KEY + "\nDELETE|COMPLETED")
print "-----"
sleep(5)

print "Видаляємо команду: "
print wc.sendmsg("ID|" + KEY +
"\nDELETE|a5051a665c837d56b21f5d612e15e9992fe68f27")
print "-----"
sleep(5)
print "Поставлено під сумнів лише ім'я файлу, %ul/dl, dlsize"
print wc.sendmsg("ID|" + KEY + "\nQUERY|filename, ratio, dlsize")
print "-----"
sleep(5)
print "Помилка запиту"
print wc.sendmsg("ID|" + KEY + "\nQUERY|filename, error, ratio")
print "-----"
sleep(5)
print "Поставлено під сумнів все"
print wc.sendmsg("ID|" + KEY +
"\nQUERY|filename, progress, btstatus, eta, dlsize, ulspeed, ratio, peers, seeds, copies
, dlsize, ulsize, peeravgprogress, totalspeed, totalsize")
print "-----"

```

CH_BitTorrent_client_option.py - файл опцій клієнта торренту

```

import sys
import wx
import os

from random import shuffle
from traceback import print_exc
from cStringIO import StringIO

from wx.lib import masked, colourselect

from CH_BitTorrent_client.GUI.menu import MenuDialog
from Utility.configreader import ConfigReader
from Utility.constants import * #IGNORE:W0611

#####
#
# Class: CH_BitTorrent_client_optionPanel
#
# Основна структура для панелі вікна опцій
#
# Додаємо кнопку для "Відновити типові значення"
# внизу кожної панелі
#
#####
class CH_BitTorrent_client_optionPanel(wx.Panel):
    def __init__(self, parent, dialog):
        wx.Panel.__init__(self, parent, -1)

        self.dialog = dialog
        self.utility = dialog.utility

        self.changed = False

        self.outersizer = wx.BoxSizer(wx.VERTICAL)

        self.sizer = wx.BoxSizer(wx.VERTICAL)

        # Речі, щоб зробити після того, як підклас закінчив свою стадію
        ініціалізації
        def initTasks(self):
            self.loadValues()

            self.outersizer.Add(self.sizer, 1, wx.EXPAND)

            defaultsButton = wx.Button(self, -1,
self.utility.lang.get('reverttodefault'))
            wx.EVT_BUTTON(self, defaultsButton.GetId(), self.setDefaults)
            self.outersizer.Add(defaultsButton, 0, wx.ALIGN_RIGHT|wx.TOP|wx.BOTTOM,
10)

            self.SetSizerAndFit(self.outersizer)

        def loadValues(self, Read = None):
            # Підроблена функція, яку члени класу повинні відкинути
            pass

        def setDefaults(self, event = None):
            self.loadValues(self.utility.config.ReadDefault)

        def apply(self):
            # Підроблена функція, яку члени класу повинні відкинути
            pass

```

```

#####
#
# Class: NetworkPanel
#
# Містить мережні установки
#
#####
class NetworkPanel(CH_BitTorrent_client_optionPanel):
    def __init__(self, parent, dialog):
        CH_BitTorrent_client_optionPanel.__init__(self, parent, dialog)
        sizer = self.sizer

        self.minport = self.utility.makeNumCtrl(self, 1, min = 1, max = 65536)
        port_box = wx.BoxSizer(wx.HORIZONTAL)
        port_box.Add(wx.StaticText(self, -1,
self.utility.lang.get('portnumber')), 0, wx.ALIGN_CENTER_VERTICAL)
        port_box.Add(self.minport, 0, wx.ALIGN_CENTER_VERTICAL|wx.LEFT|wx.RIGHT,
5)

        port_box.Add(wx.StaticText(self, -1,
self.utility.lang.get('restartCH_BitTorrent_client')), 0,
wx.ALIGN_CENTER_VERTICAL)

        sizer.Add(port_box, 0, wx.EXPAND|wx.ALL, 5)

        self.kickban = wx.CheckBox(self, -1, self.utility.lang.get('kickban'))
        sizer.Add(self.kickban, 0, wx.ALIGN_LEFT|wx.ALL, 5)

        self.notsameip = wx.CheckBox(self, -1,
self.utility.lang.get('security'))
        sizer.Add(self.notsameip, 0, wx.ALIGN_LEFT|wx.ALL, 5)

        # Робимо або не отримуємо частини завантажених даних
        #####
        self.scrape = wx.CheckBox(self, -1, self.utility.lang.get('scrape'))
        sizer.Add(self.scrape, 0, wx.ALIGN_LEFT|wx.ALL, 5)
        self.scrape.SetToolTipString(self.utility.lang.get('scrape_hint'))

        #####
        #self.ipv6 = wx.CheckBox(self, -1, "Ініціюємо і отримуємо з'єднання
IPv6")
        #if self.utility.config.Read('ipv6') == "1":
        #    self.ipv6.SetValue(True)
        #else:
        #    self.ipv6.SetValue(False)
        #####

        self.initTasks()

    def loadValues(self, Read = None):
        if Read is None:
            Read = self.utility.config.Read

        self.minport.SetValue(Read('minport', 'int'))

        self.kickban.SetValue(Read('kickban', "boolean"))
        self.notsameip.SetValue(Read('notsameip', "boolean"))
        self.scrape.SetValue(Read('scrape', "boolean"))

    def apply(self):
        minport = int(self.minport.GetValue())
        if minport > 65535:
            minport = 65535

        minchanged = self.utility.config.Write('minport', minport)

        self.utility.config.Write('kickban', self.kickban.GetValue(), "boolean")
        self.utility.config.Write('notsameip', self.notsameip.GetValue(),
"boolean")
        self.utility.config.Write('scrape', self.scrape.GetValue(), "boolean")

```

```

#####
#
# Class: QueuePanel
#
# Містить установки, які управляють, скільки потоків, щоб запустити відразу і
коли запустити їх
#
#####
class QueuePanel(CH_BitTorrent_client_optionPanel):
    def __init__(self, parent, dialog):
        CH_BitTorrent_client_optionPanel.__init__(self, parent, dialog)
        sizer = self.sizer

        #
        # Число одночасних активних потоків
        #
        activesection_title = wx.StaticBox(self, -1,
self.utility.lang.get('activeBitTorrents'))
        activesection = wx.StaticBoxSizer(activesection_title, wx.VERTICAL)

        self.numstimtext = wx.SpinCtrl(self, size = wx.Size(60, -1))
        self.numstimtext.SetRange(0, 1000)

        numsim = wx.BoxSizer(wx.HORIZONTAL)
        numsim.Add(wx.StaticText(self, -1,
self.utility.lang.get('maxnumsimul')), 0, wx.ALIGN_CENTER_VERTICAL)
        numsim.Add(self.numstimtext, 0, wx.ALIGN_CENTER_VERTICAL|wx.LEFT, 5)

        activesection.Add(numsim, 0, wx.ALL, 5)

        self.trig_finish_values = [ self.utility.lang.get('after_downloading') ,
self.utility.lang.get('after_seeding') ]
        self.trig_finish_seed = wx.ComboBox(self, -1, "", wx.Point(-1, -1),
wx.Size(-1, -1), self.trig_finish_values, wx.CB_DROPDOWN|wx.CB_READONLY)

        trigger_box = wx.BoxSizer(wx.HORIZONTAL)
        trigger_box.Add(wx.StaticText(self, -1,
self.utility.lang.get('trignextBitTorrent')), 0,
wx.ALIGN_CENTER_VERTICAL|wx.RIGHT, 5)
        trigger_box.Add(self.trig_finish_seed, 0, wx.ALIGN_CENTER_VERTICAL)

        activesection.Add(trigger_box, 0, wx.ALL, 5)

        sizer.Add(activesection, 0, wx.EXPAND|wx.ALL, 5)

        #
        # Автостарт торрентів
        #
        autostartsection_title = wx.StaticBox(self, -1,
self.utility.lang.get('autostart'))
        autostartsection = wx.StaticBoxSizer(autostartsection_title,
wx.VERTICAL)

        self.autostart = wx.CheckBox(self, -1,
self.utility.lang.get('autostart_threshold'))

        self.autostartthreshold = self.utility.makeNumCtrl(self, 0, integerWidth
= 4)

        autostart_line1box = wx.BoxSizer(wx.HORIZONTAL)
        autostart_line1box.Add(self.autostart, 0, wx.ALIGN_CENTER_VERTICAL)
        autostart_line1box.Add(self.autostartthreshold, 0,
wx.ALIGN_CENTER_VERTICAL|wx.LEFT, 5)
        autostart_line1box.Add(wx.StaticText(self, -1,
self.utility.lang.get('KB') + "/" + self.utility.lang.get('1_second')), 0,
wx.ALIGN_CENTER_VERTICAL|wx.LEFT, 3)

```

```

        self.autostartdelay = self.utility.makeNumCtrl(self, 0, integerWidth =
4, min = 1)
        autostart_line2box = wx.BoxSizer(wx.HORIZONTAL)
        autostart_line2box.Add(wx.StaticText(self, -1,
self.utility.lang.get('autostart_delay'), 0, wx.ALIGN_CENTER_VERTICAL|wx.LEFT,
20)
        autostart_line2box.Add(self.autostartdelay, 0,
wx.ALIGN_CENTER_VERTICAL|wx.LEFT, 5)
        autostart_line2box.Add(wx.StaticText(self, -1,
self.utility.lang.get('l_second')), 0, wx.ALIGN_CENTER_VERTICAL|wx.LEFT, 3)

        autostartsection.Add(autostart_line1box, 0, wx.ALL, 3)
        autostartsection.Add(autostart_line2box, 0, wx.ALL, 3)

        sizer.Add(autostartsection, 0, wx.EXPAND|wx.ALL, 5)

#
# Типове значення для нових торрентів
#
self.priorities = [ self.utility.lang.get('highest'),
                    self.utility.lang.get('high'),
                    self.utility.lang.get('normal'),
                    self.utility.lang.get('low'),
                    self.utility.lang.get('lowest') ]

        self.defaultpriority = wx.ComboBox(self, -1, "", wx.Point(-1, -1),
wx.Size(-1, -1), self.priorities, wx.CB_DROPDOWN|wx.CB_READONLY)

        prio_box = wx.BoxSizer(wx.HORIZONTAL)
        prio_box.Add(wx.StaticText(self, -1,
self.utility.lang.get('defaultpriority')), 0, wx.ALIGN_CENTER_VERTICAL|wx.RIGHT,
5)
        prio_box.Add(self.defaultpriority, 0, wx.ALIGN_CENTER_VERTICAL)
        sizer.Add(prio_box, 0, wx.ALL, 5)

        self.failbehaviors = [ self.utility.lang.get('stop'),
self.utility.lang.get('queue') ]
        self.failbehavior = wx.ComboBox(self, -1, "", wx.Point(-1, -1),
wx.Size(-1, -1), self.failbehaviors, wx.CB_DROPDOWN|wx.CB_READONLY)

        fail_box = wx.BoxSizer(wx.HORIZONTAL)
        fail_box.Add(wx.StaticText(self, -1,
self.utility.lang.get('failbehavior1')), 0, wx.ALIGN_CENTER_VERTICAL|wx.RIGHT,
5)
        fail_box.Add(self.failbehavior, 0, wx.ALIGN_CENTER_VERTICAL)
        fail_box.Add(wx.StaticText(self, -1,
self.utility.lang.get('failbehavior2')), 0, wx.ALIGN_CENTER_VERTICAL|wx.LEFT, 5)
        sizer.Add(fail_box, 0, wx.ALL, 5)

        self.fastresume = wx.CheckBox(self, -1,
self.utility.lang.get('fastresume'))

self.fastresume.SetToolTipString(self.utility.lang.get('fastresume_hint'))
        sizer.Add(self.fastresume, 0, wx.ALL, 5)

#
        self.skipcheck = wx.CheckBox(self, -1,
self.utility.lang.get('skipcheck'))
#
self.skipcheck.SetToolTipString(self.utility.lang.get('skipcheck_hint'))
#
        sizer.Add(self.skipcheck, 0, wx.ALL, 5)

        self.initTasks()

def loadValues(self, Read = None):
    if Read is None:
        Read = self.utility.config.Read

    if Read('trigwhenfinishseed', "boolean"):
        trig_default_value = self.utility.lang.get('after_seeding')

```

```

else:
    trig_default_value = self.utility.lang.get('after_downloading')
    self.trig_finish_seed.SetStringSelection(trig_default_value)

    currentprio = Read('defaultpriority', "int")
    if currentprio >= len(self.priorities):
        currentprio = len(self.priorities) - 1
    defaultprio = self.priorities[currentprio]
    self.defaultpriority.SetStringSelection(defaultprio)

    defaultfail = self.failbehaviors[Read('failbehavior', "int")]
    self.failbehavior.SetStringSelection(defaultfail)

    self.numsimtext.SetValue(Read('numsimdownload', "int"))

    self.fastresume.SetValue(Read('fastresume', "boolean"))

#     self.skipcheck.SetValue(Read('skipcheck', "boolean"))

    self.autostart.SetValue(Read('urm', "boolean"))
    self.autostartthreshold.SetValue(Read('urmupthreshold', "int"))
    self.autostartdelay.SetValue(Read('urmdelay', "int"))

    def apply(self):
        self.utility.config.Write('fastresume', self.fastresume.GetValue(),
"boolean")
#         self.utility.config.Write('skipcheck', self.skipcheck.GetValue(),
"boolean")

        selected = self.priorities.index(self.defaultpriority.GetValue())
        self.utility.config.Write('defaultpriority', selected)

        self.utility.config.Write('failbehavior',
self.failbehaviors.index(self.failbehavior.GetValue()))

        self.utility.config.Write('numsimdownload', self.numsimtext.GetValue())

        trigwhenfinished = (self.trig_finish_seed.GetValue() ==
self.utility.lang.get('after_seeding'))
        self.utility.config.Write('trigwhenfinishseed', trigwhenfinished,
"boolean")

        self.utility.config.Write('urm', self.autostart.GetValue(), "boolean")
        self.utility.config.Write('urmupthreshold',
self.autostartthreshold.GetValue())
        self.utility.config.Write('urmdelay', self.autostartdelay.GetValue())

        self.utility.queue.UpdateRunningBitTorrentCounters()

#####
#
# Class: DisplayPanel
#
# Містить установки, як виглядає CH_BitTorrent_client
#
#####
class DisplayPanel(CH_BitTorrent_client_optionPanel):
    def __init__(self, parent, dialog):
        CH_BitTorrent_client_optionPanel.__init__(self, parent, dialog)
        sizer = self.sizer

        listfont_box = wx.BoxSizer(wx.HORIZONTAL)

        listfont_box.Add(wx.StaticText(self, -1,
self.utility.lang.get('listfont')), 0, wx.ALIGN_CENTER_VERTICAL|wx.RIGHT, 5)

        self.fontexample = wx.TextCtrl(self, -1,
self.utility.lang.get('sampletext'))

```

```

listfont_box.Add(self.fontexample, 1, wx.ALIGN_CENTER_VERTICAL|wx.RIGHT,
5)

fontbutton = browsebtn = wx.Button(self, -1,
self.utility.lang.get('choosefont'), style = wx.BU_EXACTFIT)
listfont_box.Add(fontbutton, 0, wx.ALIGN_CENTER_VERTICAL|wx.RIGHT, 5)
self.Bind(wx.EVT_BUTTON, self.onFontButton, fontbutton)

sizer.Add(listfont_box, 0, wx.EXPAND|wx.ALIGN_LEFT|wx.ALL, 5)

# Смурастий список опцій
stripedlist_box = wx.BoxSizer(wx.HORIZONTAL)

self.stripedlist = wx.CheckBox(self, -1,
self.utility.lang.get('stripedlist'))
stripedlist_box.Add(self.stripedlist, 0,
wx.ALIGN_CENTER_VERTICAL|wx.RIGHT, 5)

self.stripedlist_button = colourselect.ColourSelect(self, -1, "", size =
(60, 20))

stripedlist_box.Add(self.stripedlist_button, 0,
wx.ALIGN_CENTER_VERTICAL)
sizer.Add(stripedlist_box, 0, wx.ALIGN_LEFT|wx.ALL, 5)

self.savecolumnwidth = wx.CheckBox(self, -1,
self.utility.lang.get('savecolumnwidth'))
sizer.Add(self.savecolumnwidth, 0, wx.ALL, 5)

self.contextmenu = wx.Button(self, -1,
self.utility.lang.get('customizecontextmenu') + "...")
sizer.Add(self.contextmenu, 0, wx.ALL, 5)
self.Bind(wx.EVT_BUTTON, self.onContextMenuDialog, self.contextmenu)

# self.showmenuicons = wx.CheckBox(self, -1,
self.utility.lang.get('showmenuicons'))
# sizer.Add(self.showmenuicons, 0, wx.ALL, 5)

self.initTasks()

def onContextMenuDialog(self, event = None):
dialog = MenuDialog(self, 'menu_listrightclick')
dialog.ShowModal()
dialog.Destroy()

def loadValues(self, Read = None):
if Read is None:
Read = self.utility.config.Read

self.stripedlist.SetValue(Read('stripedlist', "boolean"))

self.stripedlist_button.SetValue(Read('color_stripe', "color"))

self.savecolumnwidth.SetValue(Read('savecolumnwidth', "boolean"))

# self.showmenuicons.SetValue(Read('showmenuicons', "boolean"))

# Отримуємо інформацію про шрифт
self.fontexample.SetFont(self.utility.getFontFromInfo(Read('listfont',
"bencode-fontinfo")))

def apply(self):
self.utility.config.Write('savecolumnwidth',
self.savecolumnwidth.GetValue(), "boolean")
# self.utility.config.Write('showmenuicons',
self.showmenuicons.GetValue(), "boolean")

overallchanged = False

```

```

        changed = self.utility.config.Write('stripedlist',
self.stripedlist.GetValue(), "boolean")
        if changed:
            overallchanged = True

        # Встановлюємо колор смуг
        changed = self.utility.config.Write('color_stripe',
self.stripedlist_button.GetColour(), "color")
        if changed:
            overallchanged = True

    if overallchanged:
        for CH_BitTorrent_clientBitTorrentTemp in
self.utility.BitTorrents["all"]:
            CH_BitTorrent_clientBitTorrentTemp.updateColor()

    # Встановлюємо шрифт списку
    newfont = self.fontexample.GetFont()
    newfontinfo = self.utility.getInfoFromFont(newfont)

    fontchanged = self.utility.config.Write('listfont', newfontinfo,
"bencode-fontinfo")

    if fontchanged:
        for managedlist in self.utility.lists:
            try:
                if self.utility.lists[managedlist]:
                    managedlist.loadFont()
            except:
                pass

def onFontButton(self, event = None):
    fontdata = wx.FontData()
    fontdata.EnableEffects(False)
    fontdata.SetInitialFont(self.fontexample.GetFont())

    dlg = wx.FontDialog(self, fontdata)
    if dlg.ShowModal() == wx.ID_OK:
        data = dlg.GetFontData()

        newfont = data.GetChosenFont()
        newfontinfo = self.utility.getInfoFromFont(newfont)

        oldfontinfo = self.utility.config.Read('listfont', "bencode-
fontinfo")

        changed = False
        for attr in oldfontinfo:
            if oldfontinfo[attr] != newfontinfo[attr]:
                changed = True
                break

        if changed:
            # (TODO: May need to adjust if a large font was used)
            self.fontexample.SetFont(newfont)
            self.Layout()
            self.Refresh()

#####
#
# Class: MiscPanel
#
# Містить установки, які, здається, не відповідають добре де-небудь у іншому
місці
#
#####
class MiscPanel(CH_BitTorrent_client_optionPanel):
    def __init__(self, parent, dialog):

```

```

CH_BitTorrent_client_optionPanel.__init__(self, parent, dialog)
sizer = self.sizer

self.trayoptions = [self.utility.lang.get('showtray_never'),
                    self.utility.lang.get('showtray_min'),
                    self.utility.lang.get('showtray_always')]
self.mintray = wx.RadioButton(self,
                              -1,
                              self.utility.lang.get('showtray'),
                              wx.DefaultPosition,
                              wx.DefaultSize,
                              self.trayoptions,
                              3,
                              wx.RA_SPECIFY_COLS)
sizer.Add(self.mintray, 0, wx.ALIGN_LEFT|wx.ALL, 5)

self.confirmonclose = wx.CheckBox(self, -1,
self.utility.lang.get('confirmonexit'))
sizer.Add(self.confirmonclose, 0, wx.ALIGN_LEFT|wx.ALL, 5)

# Реєстраційна асоціація (лише має сенс під вікнами)
if (sys.platform == 'win32'):
    self.associate = wx.CheckBox(self, -1,
self.utility.lang.get('associate'))
    sizer.Add(self.associate, 0, wx.ALIGN_LEFT|wx.ALL, 5)

    #прейменовуємо торрент
    self.rtdw = wx.CheckBox(self, -1, self.utility.lang.get('rtdw'))
    self.rtdw.SetValue(self.utility.config.Read('defrentorwithdest',
"boolean"))
    sizer.Add(self.rtdw, 0, wx.ALIGN_LEFT|wx.ALL, 5)

# Опції мови
if self.utility.languages == {}:
    self.getLanguages()
    self.language_names = []
    self.language_filenames = []
    for item in self.utility.languages:
        self.language_names.append(item)
        self.language_filenames.append(self.utility.languages[item])

    self.language_choice = wx.ComboBox(self, -1, "", wx.Point(-1, -1),
wx.Size(-1, -1), self.language_names, wx.CB_DROPDOWN|wx.CB_READONLY)

    lang_box = wx.BoxSizer(wx.HORIZONTAL)
    lang_box.Add(wx.StaticText(self, -1,
self.utility.lang.get('choose_language')), 0, wx.ALIGN_CENTER_VERTICAL)
    lang_box.Add(self.language_choice, 0,
wx.ALIGN_CENTER_VERTICAL|wx.LEFT|wx.RIGHT, 5)
    lang_box.Add(wx.StaticText(self, -1,
self.utility.lang.get('restartCH_BitTorrent_client')), 0,
wx.ALIGN_CENTER_VERTICAL)
    sizer.Add(lang_box, 0, wx.ALL, 5)

self.initTasks()

def loadValues(self, Read = None):
    if Read is None:
        Read = self.utility.config.Read

    mintray = Read('mintray', "int")
    if mintray >= len(self.trayoptions):
        mintray = len(self.trayoptions) - 1
    self.mintray.SetSelection(mintray)

    self.confirmonclose.SetValue(Read('confirmonclose', "boolean"))

    self.rtdw.SetValue(Read('defrentorwithdest', "boolean"))

```

```

if (sys.platform == 'win32'):
    self.associate.SetValue(Read('associate', "boolean"))

index = self.language_filenames.index(Read('language_file'))
if not self.language_names:
    # Не повинен ніколи добиратися тут -- це означає, що не знайдено
жодних дійсних файлів мови !
    sys.stderr.write("\n НЕ ЗНАЙДЕНО ЖОДНИХ ДІЙСНИХ ФАЙЛІВ МОВИ !
Додайте дійсний файл мови\n")
    defaultlang = ""
elif (index > -1):
    defaultlang = self.language_names[index]
self.language_choice.SetStringSelection(defaultlang)

def apply(self):
    self.utility.config.Write('mintray', self.mintray.GetSelection())
    self.utility.frame.tbicon.updateIcon()

    langname_index =
self.language_names.index(self.language_choice.GetValue())
    self.utility.config.Write('language_file',
self.language_filenames[langname_index])

    self.utility.config.Write('confirmonclose',
self.confirmonclose.GetValue(), "boolean")

    self.utility.config.Write('defrentorwithdest', self.rtdw.GetValue(),
"boolean")

    self.utility.config.Write('associate', self.associate.GetValue(),
"boolean")
    self.utility.regchecker.updateRegistry(self.associate.GetValue())

def getLanguages(self):
    langpath = os.path.join(self.utility.getPath(), "lang")

    dirlist = os.listdir(langpath)
    dirlist2 = []
    for filename in dirlist:
        if (filename[-5:] == '.lang'):
            dirlist2.append(filename)
    dirlist2.sort()

    # Видаляємо user.lang зі списку
    try:
        dirlist2.remove("user.lang")
    except:
        pass

    self.utility.languages = {}

    for filename in dirlist2:
        filepath = os.path.join(langpath, filename)

        config = ConfigReader(filepath, "CH_BitTorrent_client/language")
        if config.Exists('languagename'):
            self.utility.languages[config.Read('languagename')] = filename

#####
#
# Class: DiskPanel
#
# Містить установки, пов'язані із збереженням файлів
#
#####
class DiskPanel(CH_BitTorrent_client_optionPanel):
    def __init__(self, parent, dialog):
        CH_BitTorrent_client_optionPanel.__init__(self, parent, dialog)

```

```

sizer = self.sizer

self.BitTorrentbackup = wx.CheckBox(self, -1,
self.utility.lang.get('removebackupBitTorrent'))
sizer.Add(self.BitTorrentbackup, 0, wx.ALIGN_LEFT|wx.ALL, 5)

self.defaultdir = wx.CheckBox(self, -1,
self.utility.lang.get('setdefaultfolder'))

self.dir = wx.TextCtrl(self, -1, "")
browsebtn = wx.Button(self, -1, "...", style = wx.BU_EXACTFIT)
self.Bind(wx.EVT_BUTTON, self.onBrowseDir, browsebtn)

dirbox = wx.BoxSizer(wx.HORIZONTAL)
dirbox.Add(self.defaultdir, 0, wx.ALIGN_CENTER_VERTICAL)
dirbox.Add(self.dir, 0, wx.ALIGN_CENTER_VERTICAL|wx.LEFT|wx.RIGHT, 5)
dirbox.Add(browsebtn, 0, wx.ALIGN_CENTER_VERTICAL)

sizer.Add(dirbox, 0, wx.ALIGN_LEFT|wx.ALL, 5)

self.movecompleted = wx.CheckBox(self, -1,
self.utility.lang.get('movecompleted'))

self.movedir = wx.TextCtrl(self, -1, "")
movebrowsebtn = wx.Button(self, -1, "...", style = wx.BU_EXACTFIT)
self.Bind(wx.EVT_BUTTON, self.onBrowseMoveDir, movebrowsebtn)

movedirbox = wx.BoxSizer(wx.HORIZONTAL)
movedirbox.Add(self.movecompleted, 0, wx.ALIGN_CENTER_VERTICAL)
movedirbox.Add(self.movedir, 0,
wx.ALIGN_CENTER_VERTICAL|wx.LEFT|wx.RIGHT, 5)
movedirbox.Add(movebrowsebtn, 0, wx.ALIGN_CENTER_VERTICAL)

sizer.Add(movedirbox, 0, wx.ALIGN_LEFT|wx.ALL, 5)

# self.forcenewdir = wx.CheckBox(self, -1,
self.utility.lang.get('forcenewdir'))
#
self.forcenewdir.SetToolTipString(self.utility.lang.get('forcenewdir_hint'))
#
# sizer.Add(self.forcenewdir, 0, wx.ALIGN_LEFT|wx.ALL, 5)

diskfullbox = wx.BoxSizer(wx.HORIZONTAL)
self.diskfullcheckbox = wx.CheckBox(self, -1,
self.utility.lang.get('diskfullthreshold'))
self.diskfullthreshold = self.utility.makeNumCtrl(self, 1, integerWidth
= 4)
diskfullbox.Add(self.diskfullcheckbox, 0,
wx.ALIGN_CENTER_VERTICAL|wx.RIGHT, 5)
diskfullbox.Add(self.diskfullthreshold, 0,
wx.ALIGN_CENTER_VERTICAL|wx.RIGHT, 5)
diskfullbox.Add(wx.StaticText(self, -1, self.utility.lang.get('MB')), 0,
wx.ALIGN_CENTER_VERTICAL)

sizer.Add(diskfullbox, 0, wx.ALIGN_LEFT|wx.ALL, 5)

self.initTasks()

def loadValues(self, Read = None):
if Read is None:
Read = self.utility.config.Read

self.dir.SetValue(Read('defaultfolder'))
self.BitTorrentbackup.SetValue(Read('removeBitTorrent', "boolean"))
self.defaultdir.SetValue(Read('setdefaultfolder', "boolean"))
self.movecompleted.SetValue(Read('movecompleted', "boolean"))
self.movedir.SetValue(Read('defaultmovedir'))

diskfullthreshold = Read('diskfullthreshold', "int")

```

```

        if diskfullthreshold > 0:
            self.diskfullcheckbox.SetValue(True)
            self.diskfullthreshold.SetValue(diskfullthreshold)
#         self.forcenewdir.SetValue(Read('forcenewdir', "boolean"))

    def apply(self):
        self.utility.config.Write('removeBitTorrent',
self.BitTorrentbackup.GetValue(), "boolean")

        self.utility.config.Write('setdefaultfolder',
self.defaultdir.GetValue(), "boolean")
        self.utility.config.Write('defaultfolder', self.dir.GetValue())

        self.utility.config.Write('movecompleted',
self.movecompleted.GetValue(), "boolean")
        self.utility.config.Write('defaultmovedir', self.movedir.GetValue())

        if self.diskfullcheckbox.GetValue():
            diskfullthreshold = self.diskfullthreshold.GetValue()
        else:
            diskfullthreshold = 0
        self.utility.config.Write('diskfullthreshold', diskfullthreshold)

#         self.utility.config.Write('forcenewdir', self.forcenewdir.GetValue(),
"boolean")

    def onBrowseMoveDir(self, event = None):
        dlg = wx.DirDialog(self.utility.frame,
            self.utility.lang.get('choosemovedir'),
            style = wx.DD_DEFAULT_STYLE | wx.DD_NEW_DIR_BUTTON)
        if dlg.ShowModal() == wx.ID_OK:
            self.movedir.SetValue(dlg.GetPath())
        dlg.Destroy()

    def onBrowseDir(self, event = None):
        dlg = wx.DirDialog(self.utility.frame,
            self.utility.lang.get('choosedefaultdownloadfolder'),
            style = wx.DD_DEFAULT_STYLE | wx.DD_NEW_DIR_BUTTON)
        if dlg.ShowModal() == wx.ID_OK:
            self.dir.SetValue(dlg.GetPath())
        dlg.Destroy()

#####
#
# Class: SchedulerRulePanel
#
# Містить установки, пов'язані з перервами
#
#####
class SchedulerRulePanel(CH_BitTorrent_client_optionPanel):
    def __init__(self, parent, dialog):
        CH_BitTorrent_client_optionPanel.__init__(self, parent, dialog)
        sizer = self.sizer

        # GUI діалог для встановок глобального оновлення
        #####
        sizer.Add(wx.StaticText(self, -1, self.utility.lang.get('setrule')), 0,
wx.ALL, 5)

        # Затримка для контакту трекара
        tracker_val = ['00', '5', '10', '15', '30', '45', '60', '120', '180']
#minute
        self.cb_tracker = wx.ComboBox(self, -1, "", wx.Point(-1, -1),
            wx.Size(48, -1), tracker_val,
wx.CB_DROPDOWN|wx.CB_READONLY)
        tracker_box = wx.BoxSizer(wx.HORIZONTAL)
        tracker_box.Add(wx.StaticText(self, -1,
self.utility.lang.get('timeout_tracker')), 0, wx.ALIGN_CENTER_VERTICAL)

```

```

        tracker_box.Add(self.cb_tracker, 0,
wx.ALIGN_CENTER_VERTICAL|wx.LEFT|wx.RIGHT, 5)
        tracker_box.Add(wx.StaticText(self, -1,
self.utility.lang.get('minute_long')), 0, wx.ALIGN_CENTER_VERTICAL)
        sizer.Add(tracker_box, 0, wx.ALL, 5)

        # Затримка для закачування
        download_val = ['00', '10', '20', '30', '60', '90', '120', '150', '180',
'210', '240'] #minute
        self.cb_download = wx.ComboBox(self, -1, "", wx.Point(-1, -1),
wx.Size(48, -1), download_val,
wx.CB_DROPDOWN|wx.CB_READONLY)
        download_box = wx.BoxSizer(wx.HORIZONTAL)
        download_box.Add(wx.StaticText(self, -1,
self.utility.lang.get('timeout_download')), 0, wx.ALIGN_CENTER_VERTICAL)
        download_box.Add(self.cb_download, 0,
wx.ALIGN_CENTER_VERTICAL|wx.LEFT|wx.RIGHT, 5)
        download_box.Add(wx.StaticText(self, -1,
self.utility.lang.get('minute_long')), 0, wx.ALIGN_CENTER_VERTICAL)
        sizer.Add(download_box, 0, wx.ALL, 5)

        # Затримка для сідів
        upload_val = ['00', '0.5', '1', '2', '3', '4', '5', '6', '7', '8',
'9', '10', '11', '12'] #hour
        self.cb_upload = wx.ComboBox(self, -1, "", wx.Point(-1, -1),
wx.Size(48, -1), upload_val,
wx.CB_DROPDOWN|wx.CB_READONLY)
        upload_box = wx.BoxSizer(wx.HORIZONTAL)
        upload_box.Add(wx.StaticText(self, -1,
self.utility.lang.get('timeout_upload')), 0, wx.ALIGN_CENTER_VERTICAL)
        upload_box.Add(self.cb_upload, 0,
wx.ALIGN_CENTER_VERTICAL|wx.LEFT|wx.RIGHT, 5)
        upload_box.Add(wx.StaticText(self, -1,
self.utility.lang.get('hour_long')), 0, wx.ALIGN_CENTER_VERTICAL)
        sizer.Add(upload_box, 0, wx.ALL, 5)

        self.initTasks()

    def loadValues(self, Read = None):
        if Read is None:
            Read = self.utility.config.Read

            self.cb_tracker.SetStringSelection(Read('timeouttracker'))
            self.cb_download.SetStringSelection(Read('timeoutdownload'))
            self.cb_upload.SetStringSelection(Read('timeoutupload'))

    def apply(self):
        # Встановлюємо значення затримок
        self.utility.config.Write('timeouttracker', self.cb_tracker.GetValue())
        self.utility.config.Write('timeoutdownload',
self.cb_download.GetValue())
        self.utility.config.Write('timeoutupload', self.cb_upload.GetValue())

#####
#
# Class: RateLimitPanel
#
# Містить установки, пов'язані з врегулюванням обмежень на норми роздач і
завантажень
#
#####
class RateLimitPanel(CH_BitTorrent_client_optionPanel):
    def __init__(self, parent, dialog):
        CH_BitTorrent_client_optionPanel.__init__(self, parent, dialog)
        sizer = self.sizer

        # GUI діалог для встановок глобального оновлення
        #####

```

```

# Установки роздач
#####

uploadsection_title = wx.StaticBox(self, -1,
self.utility.lang.get('uploadsetting'))
uploadsection = wx.StaticBoxSizer(uploadsection_title, wx.VERTICAL)

self.maxupload = wx.SpinCtrl(self, size = wx.Size(60, -1))
self.maxupload.SetRange(2, 100)

maxuploadsbox = wx.BoxSizer(wx.HORIZONTAL)
maxuploadsbox.Add(wx.StaticText(self, -1,
self.utility.lang.get('maxuploads')), 0, wx.ALIGN_CENTER_VERTICAL)
maxuploadsbox.Add(self.maxupload, 0, wx.ALIGN_CENTER_VERTICAL|wx.LEFT,
5)

uploadsection.Add(maxuploadsbox, 0, wx.ALIGN_CENTER_VERTICAL|wx.ALL, 5)

maxoverall_down_label = wx.BoxSizer(wx.VERTICAL)
maxoverall_down_label.Add(wx.StaticText(self, -1,
self.utility.lang.get('maxoveralluploadrate')), 0, wx.ALIGN_CENTER_VERTICAL)
maxoverall_down_label.Add(wx.StaticText(self, -1,
self.utility.lang.get('whendownload')), 0, wx.ALIGN_CENTER_VERTICAL)

self.uploadrate = self.utility.makeNumCtrl(self, 0, integerWidth = 4)
self.uploadrate.SetToolTipString(self.utility.lang.get('global_uprate_hint'))

maxoverall_down = wx.BoxSizer(wx.HORIZONTAL)
maxoverall_down.Add(maxoverall_down_label, 0, wx.ALIGN_CENTER_VERTICAL)
maxoverall_down.Add(self.uploadrate, 0,
wx.ALIGN_CENTER_VERTICAL|wx.LEFT, 5)
maxoverall_down.Add(wx.StaticText(self, -1, self.utility.lang.get('KB')
+ "/" + self.utility.lang.get('l_second')), 0, wx.ALIGN_CENTER_VERTICAL|wx.LEFT,
3)

uploadsection.Add(maxoverall_down, 0, wx.ALIGN_CENTER_VERTICAL|wx.ALL,
5)

maxoverall_nodown_label = wx.BoxSizer(wx.VERTICAL)
maxoverall_nodown_label.Add(wx.StaticText(self, -1,
self.utility.lang.get('maxoveralluploadrate')), 0, wx.ALIGN_CENTER_VERTICAL)
maxoverall_nodown_label.Add(wx.StaticText(self, -1,
self.utility.lang.get('whennodownload')), 0, wx.ALIGN_CENTER_VERTICAL)

self.seeduploadrate = self.utility.makeNumCtrl(self, 0, integerWidth =
4)
self.seeduploadrate.SetToolTipString(self.utility.lang.get('global_uprate_hint')
)

maxoverall_nodown = wx.BoxSizer(wx.HORIZONTAL)
maxoverall_nodown.Add(maxoverall_nodown_label, 0,
wx.ALIGN_CENTER_VERTICAL)
maxoverall_nodown.Add(self.seeduploadrate, 0,
wx.ALIGN_CENTER_VERTICAL|wx.LEFT, 5)
maxoverall_nodown.Add(wx.StaticText(self, -1,
self.utility.lang.get('KB') + "/" + self.utility.lang.get('l_second')), 0,
wx.ALIGN_CENTER_VERTICAL|wx.LEFT, 3)

uploadsection.Add(maxoverall_nodown, 0, wx.ALIGN_CENTER_VERTICAL|wx.ALL,
5)

uploadsection.Add(wx.StaticText(self, -1,
self.utility.lang.get('zeroisunlimited')), 0,
wx.ALIGN_CENTER_VERTICAL|wx.ALIGN_RIGHT|wx.ALL, 5)

sizer.Add(uploadsection, 0, wx.EXPAND|wx.ALL, 5)

```

```

# Завантаження по частинам
downloadsection_title = wx.StaticBox(self, -1,
self.utility.lang.get('downloadsetting'))
downloadsection = wx.StaticBoxSizer(downloadsection_title, wx.VERTICAL)

self.downloadrate = self.utility.makeNumCtrl(self, 0, integerWidth = 4)

maxdownoverall_down = wx.BoxSizer(wx.HORIZONTAL)
maxdownoverall_down.Add(wx.StaticText(self, -1,
self.utility.lang.get('maxoveralldownloadrate')), 0, wx.ALIGN_CENTER_VERTICAL)
maxdownoverall_down.Add(self.downloadrate, 0,
wx.ALIGN_CENTER_VERTICAL|wx.LEFT, 5)
maxdownoverall_down.Add(wx.StaticText(self, -1,
self.utility.lang.get('KB') + "/" + self.utility.lang.get('1_second')), 0,
wx.ALIGN_CENTER_VERTICAL|wx.LEFT, 3)

downloadsection.Add(maxdownoverall_down, 0,
wx.ALIGN_CENTER_VERTICAL|wx.ALL, 5)

downloadsection.Add(wx.StaticText(self, -1,
self.utility.lang.get('zeroisunlimited')), 0,
wx.ALIGN_CENTER_VERTICAL|wx.ALIGN_RIGHT|wx.ALL, 5)

sizer.Add(downloadsection, 0, wx.EXPAND|wx.ALL, 5)

# Встановлення локальних пріоритетів
self.prioritize_local = wx.CheckBox(self, -1,
self.utility.lang.get('prioritize_local'))
sizer.Add(self.prioritize_local, 0, wx.ALL, 5)

self.initTasks()

def loadValues(self, Read = None):
if Read is None:
Read = self.utility.config.Read

self.maxupload.SetValue(Read('maxupload', "int"))
self.uploadrate.SetValue(Read('maxuploadrate', "int"))
self.downloadrate.SetValue(Read('maxdownloadrate', "int"))
self.seeduploadrate.SetValue(Read('maxseeduploadrate', "int"))
self.prioritize_local.SetValue(Read('prioritize_local', "boolean"))

def apply(self):
# Перевірка максимального рівня роздачі, повинна бути цілою
#####
upload_rate = int(self.uploadrate.GetValue())
seedupload_rate = int(self.seeduploadrate.GetValue())

download_rate = int(self.downloadrate.GetValue())

# Перевірка максимального рівня роздачі, повинна бути не менш 3 kB/s
#####
if (upload_rate < 3 and upload_rate != 0) or (seedupload_rate < 3 and
seedupload_rate != 0):
#виведення попереджень на екран
dlg = wx.MessageDialog(self,
self.utility.lang.get('uploadrateminwarning'), self.utility.lang.get('error'),
wx.ICON_ERROR)
dlg.ShowModal()
dlg.Destroy()
return False

self.utility.config.Write('prioritize_local',
self.prioritize_local.GetValue(), "boolean")

# Встановлення нових значень параметрів
#####
self.utility.config.Write('maxupload', self.maxupload.GetValue())

```

```

self.utility.config.Write('maxuploadrate', upload_rate)
self.utility.config.Write('maxseeduploadrate', seedupload_rate)

self.utility.config.Write('maxdownloadrate', download_rate)

#####
#
# Class: SeedingOptionsPanel
#
# Містить опції, які управляють довжиною торрентів.
#
#####
class SeedingOptionsPanel(CH_BitTorrent_client_optionPanel):
    def __init__(self, parent, dialog):
        CH_BitTorrent_client_optionPanel.__init__(self, parent, dialog)
        sizer = self.Sizer

        # GUI діалог для встановок глобального оновлення
        #####

        # Установки роздачі для повних файлів
        #####

        continuesection_title = wx.StaticBox(self, -1,
self.utility.lang.get('uploadoptforcompletedfile'))
        continuesection = wx.StaticBoxSizer(continuesection_title, wx.VERTICAL)

        uploadlist = [self.utility.lang.get('unlimitedupload'),
self.utility.lang.get('continueuploadfor'), self.utility.lang.get('untilratio')]

        rb1 = wx.RadioButton(self, -1, uploadlist[0], wx.Point(-1, -1),
wx.Size(-1, -1), wx.RB_GROUP)
        rb2 = wx.RadioButton(self, -1, uploadlist[1], wx.Point(-1, -1),
wx.Size(-1, -1))
        rb3 = wx.RadioButton(self, -1, uploadlist[2], wx.Point(-1, -1),
wx.Size(-1, -1))
        self.rb = [rb1, rb2, rb3]

        mtimeval = ['30', '45', '60', '75']
        htimeval = []
        for i in range(24):
            htimeval.append(str(i))

        self.cbhtime = wx.ComboBox(self, -1, "", wx.Point(-1, -1),
wx.Size(37, -1), htimeval,
wx.CB_DROPDOWN|wx.CB_READONLY)
        self.cbmtime = wx.ComboBox(self, -1, "", wx.Point(-1, -1),
wx.Size(37, -1), mtimeval,
wx.CB_DROPDOWN|wx.CB_READONLY)

        continuesection.Add(rb1, 0, wx.ALIGN_CENTER_VERTICAL|wx.ALL, 5)

        time_sizer = wx.BoxSizer(wx.HORIZONTAL)
        time_sizer.Add(rb2, 0, wx.ALIGN_CENTER_VERTICAL)
        time_sizer.Add(self.cbhtime, 0,
wx.ALIGN_CENTER_VERTICAL|wx.LEFT|wx.RIGHT, 5)
        time_sizer.Add(wx.StaticText(self, -1, self.utility.lang.get('hour')),
0, wx.ALIGN_CENTER_VERTICAL)
        time_sizer.Add(self.cbmtime, 0,
wx.ALIGN_CENTER_VERTICAL|wx.LEFT|wx.RIGHT, 5)
        time_sizer.Add(wx.StaticText(self, -1, self.utility.lang.get('minute')),
0, wx.ALIGN_CENTER_VERTICAL)

        continuesection.Add(time_sizer, -1, wx.ALIGN_CENTER_VERTICAL|wx.ALL, 5)

        ratioval = ['50', '75', '100', '125', '150', '175', '200', '300', '400',
'500']
        self.cbratio = wx.ComboBox(self, -1, "",

```

```

wx.Point(-1, -1), wx.Size(45, -1), ratioval,
wx.CB_DROPDOWN|wx.CB_READONLY)

    percent_sizer = wx.BoxSizer(wx.HORIZONTAL)
    percent_sizer.Add(rb3, 0, wx.ALIGN_CENTER_VERTICAL)
    percent_sizer.Add(self.cb_ratio, 0, wx.ALIGN_CENTER_VERTICAL|wx.LEFT, 5)
    percent_sizer.Add(wx.StaticText(self, -1, "%"), 0,
wx.ALIGN_CENTER_VERTICAL)

    continuesection.Add(percent_sizer, 0, wx.ALIGN_CENTER_VERTICAL|wx.ALL,
5)

    sizer.Add(continuesection, 0, wx.EXPAND|wx.ALL, 5)

    self.initTasks()

def loadValues(self, Read = None):
    if Read is None:
        Read = self.utility.config.Read

    self.cb_ratio.SetValue(Read('uploadratio'))
    self.cb_h_time.SetValue(Read('uploadtimeh'))
    self.cb_m_time.SetValue(Read('uploadtimem'))

    self.rb[Read('uploadoption', "int")].SetValue(True)

def apply(self):
    # Встановлення нових значень параметрів
    #####
    for i in range(3):
        if self.rb[i].GetValue():
            self.utility.config.Write('uploadoption', i)

    self.utility.config.Write('uploadtimeh', self.cb_h_time.GetValue())
    self.utility.config.Write('uploadtimem', self.cb_m_time.GetValue())
    self.utility.config.Write('uploadratio', self.cb_ratio.GetValue())

#####
#
# Class: ColorPanel
#
# Містить опції, які міняють кольор для різних торрентів
#
#####
class ColorPanel(CH_BitTorrent_client_optionPanel):
    def __init__(self, parent, dialog):
        CH_BitTorrent_client_optionPanel.__init__(self, parent, dialog)
        sizer = self.sizer

        self.colors = ['color_startup',
                        'color_disconnected',
                        'color_noconnections',
                        'color_noincoming',
                        'color_nocomplete',
                        'color_good' ]

        color_boxes = {}
        color_text = {}
        self.color_buttons = {}

        # Смурастий список опцій
        for color in self.colors:
            color_boxes[color] = wx.BoxSizer(wx.HORIZONTAL)

            self.color_buttons[color] = colourselect.ColourSelect(self, -1, "",
size = (60, 20))
            color_boxes[color].Add(self.color_buttons[color], 0,
wx.ALIGN_CENTER_VERTICAL|wx.RIGHT, 5)

```

```

        color_text[color] = wx.StaticText(self, -1,
self.utility.lang.get(color))
        color_boxes[color].Add(color_text[color], 0,
wx.ALIGN_CENTER_VERTICAL)

        sizer.Add(color_boxes[color], 0, wx.ALIGN_LEFT|wx.ALL, 5)

self.initTasks()

def loadValues(self, Read = None):
    if Read is None:
        Read = self.utility.config.Read

    for color in self.colors:
        self.color_buttons[color].SetValue(Read(color, "color"))

def apply(self):
    overallchange = False

    for color in self.colors:
        color_value = self.color_buttons[color].GetColour()
        changed = self.utility.config.Write(color, color_value, "color")
        if changed:
            overallchange = True

    if overallchange:
        for CH_BitTorrent_clientBitTorrentTemp in
self.utility.BitTorrents["all"]:
            CH_BitTorrent_clientBitTorrentTemp.updateColor()

#####
#
# Class: AdvancedNetworkPanel
#
# Містить різні мережні встановлення (типові значення мають бути оптимальні для
більшості користувачів)
#
#####
class AdvancedNetworkPanel(CH_BitTorrent_client_optionPanel):
    def __init__(self, parent, dialog):
        CH_BitTorrent_client_optionPanel.__init__(self, parent, dialog)
        sizer = self.sizer

        warningtext = wx.StaticText(self, -1,
self.utility.lang.get('changeownrisk'))
        sizer.Add(warningtext, 0, wx.ALIGN_CENTER|wx.ALL, 5)

        #self.ipv6bindsv4_data = wx.Choice(self, -1,
        #    choices = ['separate sockets', 'single socket'])

#self.ipv6bindsv4_data.SetSelection(int(self.advancedConfig['ipv6_binds_v4']))

        datasizer = wx.FlexGridSizer(cols = 2, vgap = 5, hgap = 10)

        # Локальний IP
        self.ip_data = wx.TextCtrl(self, -1)
        datasizer.Add(wx.StaticText(self, -1, self.utility.lang.get('localip')),
1, wx.ALIGN_CENTER_VERTICAL)
        datasizer.Add(self.ip_data)

        # IP для з'єднання
        self.bind_data = wx.TextCtrl(self, -1)
        datasizer.Add(wx.StaticText(self, -1,
self.utility.lang.get('iptobindto')), 1, wx.ALIGN_CENTER_VERTICAL)
        datasizer.Add(self.bind_data)

        # Мінімальна кількість пірів
        self.minpeers_data = wx.SpinCtrl(self, size = wx.Size(60, -1))

```

```

        self.minpeers_data.SetRange(10, 100)
        datasizer.Add(wx.StaticText(self, -1,
self.utility.lang.get('minnumberofpeer')), 1, wx.ALIGN_CENTER_VERTICAL)
        datasizer.Add(self.minpeers_data)

        # Максимальне число з'єднань
        self.maxconnections_choices = [self.utility.lang.get('nolimit'), '20',
'30', '40', '60', '100', '200']
        self.maxconnections_data = wx.Choice(self, -1, wx.Point(-1, -1),
wx.Size(-1, -1), self.maxconnections_choices)
        datasizer.Add(wx.StaticText(self, -1,
self.utility.lang.get('maxpeerconnection')), 1, wx.ALIGN_CENTER_VERTICAL)
        datasizer.Add(self.maxconnections_data)

        # UPnP установки
        self.upnp_choices = [ self.utility.lang.get('upnp_0'),
self.utility.lang.get('upnp_1'),
self.utility.lang.get('upnp_2') ]
        self.upnp_data = wx.ComboBox(self, -1, "", wx.Point(-1, -1), wx.Size(-1,
-1), self.upnp_choices, wx.CB_DROPDOWN|wx.CB_READONLY)
        datasizer.Add(wx.StaticText(self, -1, self.utility.lang.get('upnp')), 1,
wx.ALIGN_CENTER_VERTICAL)
        datasizer.Add(self.upnp_data)

        sizer.Add(datasizer, 0, wx.ALL, 5)

        # Встановлюємо контекстні вікна покажчика
        self.ip_data.SetToolTipString(self.utility.lang.get('iphint'))
        self.bind_data.SetToolTipString(self.utility.lang.get('bindhint'))

self.minpeers_data.SetToolTipString(self.utility.lang.get('minpeershint'))

self.maxconnections_data.SetToolTipString(self.utility.lang.get('maxconnectionhi
nt'))

        self.initTasks()

def loadValues(self, Read = None):
    if Read is None:
        Read = self.utility.config.Read

        self.ip_data.SetValue(Read('ip'))
        self.bind_data.SetValue(Read('bind'))

        self.minpeers_data.SetValue(Read('min_peers', "int"))

        setval = Read('max_connections', "int")
        if setval == 0:
            setval = self.utility.lang.get('nolimit')
        else:
            setval = str(setval)
        if not setval in self.maxconnections_choices:
            setval = self.maxconnections_choices[0]
        self.maxconnections_data.SetStringSelection(setval)

        upnp_val = self.utility.config.Read('upnp_nat_access', "int")
        if upnp_val >= len(self.upnp_choices):
            upnp_val = len(self.upnp_choices) - 1
        self.upnp_data.SetStringSelection(self.upnp_choices[upnp_val])

#         #self.ipv6bindsv4_data.SetSelection()

def apply(self):
    #if self.ipv6.GetValue():
    #     self.utility.config.Write('ipv6') = "1"
    #else:
    #     self.utility.config.Write('ipv6') = "0"
    self.utility.config.Write('ipv6', "0")

```

```

        # Розширені опції
self.utility.config.Write('ip', self.ip_data.GetValue())
self.utility.config.Write('bind', self.bind_data.GetValue())

minpeers = self.minpeers_data.GetValue()
self.utility.config.Write('min_peers', minpeers)

try:
    maxconnections = int(self.maxconnections_data.GetStringSelection())
    maxinitiate = min(2 * minpeers, maxconnections)
except:      # Якщо немає числа, це повинно бути "безлімітним"
    maxconnections = 0
    maxinitiate = 2 * minpeers

self.utility.config.Write('max_initiate', maxinitiate)
self.utility.config.Write('max_connections', maxconnections)

upnp_choices = [ self.utility.lang.get('upnp_0'),
                  self.utility.lang.get('upnp_1'),
                  self.utility.lang.get('upnp_2') ]
selected = upnp_choices.index(self.upnp_data.GetValue())
self.utility.config.Write('upnp_nat_access', selected)

self.utility.config.Write('ipv6_binds_v4', "1")

#####
#
# Class: AdvancedDiskPanel
#
# Містяться розширені встановки, як дані записуються на диск, й читаються з
нього
# (повинно бути оптимальним для більшості користувачів)
#
#####
class AdvancedDiskPanel(CH_BitTorrent_client_optionPanel):
    def __init__(self, parent, dialog):
        CH_BitTorrent_client_optionPanel.__init__(self, parent, dialog)
        sizer = self.sizer

        warningtext = wx.StaticText(self, -1,
self.utility.lang.get('changeownrisk'))
        sizer.Add(warningtext, 0, wx.ALIGN_CENTER|wx.ALL, 5)

        datasizer = wx.FlexGridSizer(cols = 2, vgap = 5, hgap = 10)

        # Allocation Type

        alloc_choices = [self.utility.lang.get('alloc_normal'),
                        self.utility.lang.get('alloc_background'),
                        self.utility.lang.get('alloc_prealloc'),
                        self.utility.lang.get('alloc_sparse')]
        self.alloc_strings = {"normal": 0, "background": 1, "pre-allocate": 2,
"sparse": 3}
        self.alloctype_data = wx.Choice(self, -1, wx.Point(-1, -1), wx.Size(-1,
-1), alloc_choices)

        datasizer.Add(wx.StaticText(self, -1,
self.utility.lang.get('diskalloctype')), 1, wx.ALIGN_CENTER_VERTICAL)
        datasizer.Add(self.alloctype_data)

        # Стандартне розміщення
        self.allocrate_data = wx.SpinCtrl(self, size = wx.Size(60, -1))
        self.allocrate_data.SetRange(1, 100)
        datasizer.Add(wx.StaticText(self, -1,
self.utility.lang.get('allocrate')), 1, wx.ALIGN_CENTER_VERTICAL)

        allocrate_box = wx.BoxSizer(wx.HORIZONTAL)
        allocrate_box.Add(self.allocrate_data)

```

```

        allocate_box.Add(wx.StaticText(self, -1, " " +
self.utility.lang.get('mb') + "/" + self.utility.lang.get("l_second")), 1,
wx.ALIGN_CENTER_VERTICAL)

        datasizer.Add(allocate_box)

        # Метод блокування
        locking_choices = [self.utility.lang.get('lock_never'),
                            self.utility.lang.get('lock_writing'),
                            self.utility.lang.get('lock_always')]
        self.locking_data = wx.Choice(self, -1, wx.Point(-1, -1), wx.Size(-1, -
1), locking_choices)
        datasizer.Add(wx.StaticText(self, -1,
self.utility.lang.get('filelocking')), 1, wx.ALIGN_CENTER_VERTICAL)
        datasizer.Add(self.locking_data)

        # Метод двійної перевірки
        doublecheck_choices = [self.utility.lang.get('check_none'),
                                self.utility.lang.get('check_double'),
                                self.utility.lang.get('check_triple')]
        self.doublecheck_data = wx.Choice(self, -1, wx.Point(-1, -1), wx.Size(-
1, -1), doublecheck_choices)
        datasizer.Add(wx.StaticText(self, -1,
self.utility.lang.get('extradatachecking')), 1, wx.ALIGN_CENTER_VERTICAL)
        datasizer.Add(self.doublecheck_data)

        # Максимальна кількість відкритих файлів
        self.maxfilesopen_choices = ['50', '100', '200',
self.utility.lang.get('nolimit')]
        self.maxfilesopen_data = wx.Choice(self, -1, wx.Point(-1, -1), wx.Size(-
1, -1), self.maxfilesopen_choices)

        datasizer.Add(wx.StaticText(self, -1,
self.utility.lang.get('maxfileopen')), 1, wx.ALIGN_CENTER_VERTICAL)
        datasizer.Add(self.maxfilesopen_data)

        # Основні дані
        self.flush_data_enable = wx.CheckBox(self, -1,
self.utility.lang.get('flush_data'))

        self.flush_data = wx.SpinCtrl(self, size = wx.Size(60, -1))
        self.flush_data.SetRange(0, 999)

        datasizer.Add(self.flush_data_enable, 0, wx.ALIGN_CENTER_VERTICAL)

        flush_box = wx.BoxSizer(wx.HORIZONTAL)
        flush_box.Add(self.flush_data, 0, wx.ALIGN_CENTER_VERTICAL)
        flush_box.Add(wx.StaticText(self, -1,
self.utility.lang.get('minute_long')), 0, wx.ALIGN_CENTER_VERTICAL|wx.LEFT, 5)

        datasizer.Add(flush_box)

        sizer.Add(datasizer, 0, wx.ALL, 5)

        # Місце буферизації
        buffer_title = wx.StaticBox(self, -1,
self.utility.lang.get('bufferdisk'))
        buffer = wx.StaticBoxSizer(buffer_title, wx.VERTICAL)

        self.buffer_read_enable = wx.CheckBox(self, -1,
self.utility.lang.get('buffer_read'))

        self.buffer_write = wx.SpinCtrl(self, size = wx.Size(60, -1))
        self.buffer_write.SetRange(0, 999)

        self.buffer_write_enable = wx.CheckBox(self, -1,
self.utility.lang.get('buffer_write'))

        buffer_write_box = wx.BoxSizer(wx.HORIZONTAL)

```

```

        buffer_write_box.Add(self.buffer_write_enable, 0,
wx.ALIGN_CENTER_VERTICAL)
        buffer_write_box.Add(self.buffer_write, 0,
wx.ALIGN_CENTER_VERTICAL|wx.LEFT, 5)
        buffer_write_box.Add(wx.StaticText(self, -1,
self.utility.lang.get('mb')), 0, wx.ALIGN_CENTER_VERTICAL|wx.LEFT, 5)

        buffer.Add(self.buffer_read_enable, 0, wx.ALL, 5)
        buffer.Add(buffer_write_box, 0, wx.ALL, 5)

        sizer.Add(buffer, 0, wx.EXPAND|wx.ALL, 5)

self.alloctype_data.SetToolTipString(self.utility.lang.get('alloctypehint'))
self.allocrate_data.SetToolTipString(self.utility.lang.get('allocratehint'))
self.locking_data.SetToolTipString(self.utility.lang.get('lockinghint'))
self.doublecheck_data.SetToolTipString(self.utility.lang.get('doublecheckhint'))
self.maxfilesopen_data.SetToolTipString(self.utility.lang.get('maxfileopenhint')
)

    self.initTasks()

def loadValues(self, Read = None):
    if Read is None:
        Read = self.utility.config.Read

    try:
        alloc_selection = self.alloc_strings[Read('alloc_type')]
    except:
        alloc_selection = 0
    self.alloctype_data.SetSelection(alloc_selection)

    self.allocrate_data.SetValue(Read('alloc_rate', "int"))

    if Read('lock_files', "int"):
        if Read('lock_while_reading', "int"):
            self.locking_data.SetSelection(2)
        else:
            self.locking_data.SetSelection(1)
    else:
        self.locking_data.SetSelection(0)

    if Read('double_check', "int"):
        if Read('triple_check', "int"):
            self.doublecheck_data.SetSelection(2)
        else:
            self.doublecheck_data.SetSelection(1)
    else:
        self.doublecheck_data.SetSelection(0)

    setval = Read('max_files_open', "int")
    if setval == 0:
        setval = self.utility.lang.get('nolimit')
    else:
        setval = str(setval)
    if not setval in self.maxfilesopen_choices:
        setval = self.maxfilesopen_choices[0]
    self.maxfilesopen_data.SetStringSelection(setval)

    self.buffer_read_enable.SetValue(Read('buffer_read', "boolean"))

    try:
        flushval = Read('auto_flush', "int")
    except:
        flushval = 0
    self.flush_data.SetValue(flushval)

```

```

self.flush_data_enable.SetValue(flushval > 0)

try:
    writeval = Read('buffer_write', "int")
except:
    writeval = 0
self.buffer_write.SetValue(writeval)
self.buffer_write_enable.SetValue(writeval > 0)

def apply(self):
    truth = { True: "1", False: "0" }

    alloc_strings = ["normal", "background", "pre-allocate", "sparse"]

    self.utility.config.Write('alloc_type',
alloc_strings[self.alloctype_data.GetSelection()])
    self.utility.config.Write('alloc_rate',
int(self.allocrate_data.GetValue()))

    try:
        maxopen = int(self.maxfilesopen_data.GetStringSelection())
    except:
        # if it ain't a number, it must be "no limit"
        maxopen = 0
    self.utility.config.Write('max_files_open', maxopen)

    self.utility.config.Write('lock_files', self.locking_data.GetSelection()
> = 1, "boolean")
    self.utility.config.Write('lock_while_reading',
self.locking_data.GetSelection() > 1, "boolean")

    self.utility.config.Write('double_check',
self.doublecheck_data.GetSelection() > = 1, "boolean")
    self.utility.config.Write('triple_check',
self.doublecheck_data.GetSelection() > 1, "boolean")

    self.utility.config.Write('buffer_read',
self.buffer_read_enable.GetValue(), "boolean")

    if not self.buffer_write_enable.GetValue():
        writeval = 0
    else:
        writeval = self.buffer_write.GetValue()
    self.utility.config.Write('buffer_write', writeval)

    if not self.flush_data_enable.GetValue():
        flushval = 0
    else:
        flushval = self.flush_data.GetValue()
    self.utility.config.Write('auto_flush', flushval)

#####
#
# Class: CH_BitTorrent_clientTree
#
# Розмежованне внесення до всіх панелей опцій
#
#####
class CH_BitTorrent_clientTree(wx.TreeCtrl):
    def __init__(self, parent, dialog):
        style = wx.TR_DEFAULT_STYLE | wx.TR_HIDE_ROOT
        wx.TreeCtrl.__init__(self, parent, -1, style = style)

        self.dialog = dialog
        self.utility = dialog.utility

        self.root = self.AddRoot("Preferences")

```

```

        self.ratelimits = self.AppendItem(self.root,
self.utility.lang.get('ratelimits'))
        self.seedingoptions = self.AppendItem(self.root,
self.utility.lang.get('seedoptions'))
        self.queuesetting = self.AppendItem(self.root,
self.utility.lang.get('queuesetting'))
        self.timeout = self.AppendItem(self.root,
self.utility.lang.get('timeout'))

        self.network = self.AppendItem(self.root,
self.utility.lang.get('networksetting'))

        self.advancednetwork = self.AppendItem(self.network,
self.utility.lang.get('advanced'))

        self.disk = self.AppendItem(self.root,
self.utility.lang.get('disksettings'))
        self.advanceddisk = self.AppendItem(self.disk,
self.utility.lang.get('advanced'))

        self.display = self.AppendItem(self.root,
self.utility.lang.get('displaysetting'))

        self.colors = self.AppendItem(self.display,
self.utility.lang.get('BitTorrentcolors'))

        self.misc = self.AppendItem(self.root,
self.utility.lang.get('miscsetting'))

        self.treeMap = {self.ratelimits : self.dialog.rateLimitPanel,
                        self.seedingoptions : self.dialog.seedingOptionsPanel,
                        self.queuesetting : self.dialog.queuePanel,
                        self.timeout : self.dialog.schedulerRulePanel,
                        self.network : self.dialog.networkPanel,
                        self.misc : self.dialog.miscPanel,
                        self.display : self.dialog.displayPanel,
                        self.colors : self.dialog.colorPanel,
                        self.disk : self.dialog.diskPanel }

        self.treeMap[self.advancednetwork] = self.dialog.advancedNetworkPanel
        self.treeMap[self.advanceddisk] = self.dialog.advancedDiskPanel

        self.Bind(wx.EVT_TREE_SEL_CHANGED, self.onSwitchPage)

        self.SetAutoLayout(True)
        self.Fit()

def onSwitchPage(self, event = None):
    if self.dialog.closing or event is None:
        return

    newitem = event.GetItem()
    newpanel = None
    foundnew = False
    for key in self.treeMap:
        if key == newitem:
            newpanel = self.treeMap[key]
            foundnew = True
        if foundnew:
            break

    if newpanel is not None:
        # Спроба переключитися в поточне вікно
        try:
            oldpanel = self.dialog.splitter.GetWindow2()
            if oldpanel != newpanel:
                oldpanel.Show(False)
                self.dialog.splitter.ReplaceWindow(oldpanel, newpanel)
                newpanel.Show(True)

```

```

        newpanel.changed = True
    except:
        pass

#####
#
# Class: CH_BitTorrent_client_optionDialog
#
# Створює діалог, який дозволяє користувачам встановлювати різні переваги
#
#####
class CH_BitTorrent_client_optionDialog(wx.Dialog):
    def __init__(self, parent):
        self.utility = parent.utility

        style = wx.DEFAULT_DIALOG_STYLE | wx.RESIZE_BORDER
        # size = wx.Size(530, 420)

        size, split = self.getWindowSettings()

        wx.Dialog.__init__(self, parent, -1,
self.utility.lang.get('CH_BitTorrent_clientpreference'), size = size, style =
style)

        self.splitter = wx.SplitterWindow(self, -1, style = wx.SP_NOBORDER |
wx.SP_LIVE_UPDATE)

        self.rateLimitPanel = RateLimitPanel(self.splitter, self)
        self.seedingOptionsPanel = SeedingOptionsPanel(self.splitter, self)
        self.queuePanel = QueuePanel(self.splitter, self)

        self.schedulerRulePanel = SchedulerRulePanel(self.splitter, self)
        self.networkPanel = NetworkPanel(self.splitter, self)
        self.miscPanel = MiscPanel(self.splitter, self)
        self.displayPanel = DisplayPanel(self.splitter, self)
        self.colorPanel = ColorPanel(self.splitter, self)
        self.diskPanel = DiskPanel(self.splitter, self)

        self.advancedNetworkPanel = AdvancedNetworkPanel(self.splitter, self)
        self.advancedDiskPanel = AdvancedDiskPanel(self.splitter, self)

        self.tree = CH_BitTorrent_clientTree(self.splitter, self)

        self.splitter.SetAutoLayout(True)
        self.splitter.Fit()

        applybtn = wx.Button(self, -1, "
"+self.utility.lang.get('apply')+ " ", size = (60, -1))
        okbtn = wx.Button(self, -1, " "+self.utility.lang.get('ok')+
", size = (60, -1))
        cancelbtn = wx.Button(self, -1, "
"+self.utility.lang.get('cancel')+ " ", size = (60, -1))

        buttonbox = wx.BoxSizer(wx.HORIZONTAL)
        buttonbox.Add(applybtn, 0, wx.ALL, 5)
        buttonbox.Add(okbtn, 0, wx.ALL, 5)
        buttonbox.Add(cancelbtn, 0, wx.ALL, 5)

        outerbox = wx.BoxSizer(wx.VERTICAL)
        outerbox.Add(self.splitter , 1, wx.EXPAND|wx.LEFT|wx.RIGHT|wx.TOP, 5)

        outerbox.Add(wx.StaticLine(self, -1), 0,
wx.EXPAND|wx.LEFT|wx.RIGHT|wx.BOTTOM, 5)

        outerbox.Add(buttonbox, 0, wx.ALIGN_RIGHT)

        # Додає події
        #####
        self.Bind(wx.EVT_BUTTON, self.onOK, okbtn)

```

```

self.Bind(wx.EVT_BUTTON, self.onApply, applybtn)
self.Bind(wx.EVT_BUTTON, self.onCloseGlobalPref, cancelbtn)
self.Bind(wx.EVT_CLOSE, self.onCloseGlobalPref)

self.splitter.SplitVertically(self.tree, self.rateLimitPanel, split)
self.rateLimitPanel.changed = True
self.splitter.SetMinimumPaneSize(50)

for key in self.tree.treeMap:
    panel = self.tree.treeMap[key]
    panel.Show(False)

self.rateLimitPanel.Show(True)
self.rateLimitPanel.Fit()

self.SetSizer(outerbox)
# self.Fit()

self.closing = False

def getWindowSettings(self):
    width = self.utility.config.Read("prefwindow_width", "int")
    height = self.utility.config.Read("prefwindow_height", "int")
    split = self.utility.config.Read("prefwindow_split", "int")

    return wx.Size(width, height), split

def saveWindowSettings(self):
    width, height = self.GetSizeTuple()
    self.utility.config.Write("prefwindow_width", width)
    self.utility.config.Write("prefwindow_height", height)
    self.utility.config.Write("prefwindow_split",
self.splitter.GetSashPosition())
    self.utility.config.Flush()

def onCloseGlobalPref(self, event = None):
    self.closing = True

    self.saveWindowSettings()

    self.EndModal(wx.ID_CANCEL)

def onApply(self, event = None):
    # Встановлення нових значень параметрів
    #####

    # Лише застосовуємо зміни для груп, які розглянув користувач
    for key in self.tree.treeMap:
        panel = self.tree.treeMap[key]
        if panel.changed:
            panel.apply()

    # записуємо поточні зміни на диск
    self.utility.config.Flush()

    self.utility.queue.changeCH_BitTorrent_clientParams() #прапор
перезапису

    return True

def onOK(self, event = None):
    if self.onApply():
        self.closing = True
        self.saveWindowSettings()

        self.EndModal(wx.ID_OK)

```

utility.py - додаткові утиліти

```

import wx
import sys
import os
import socket
import random

from Lang.lang import Lang
from threading import Event, Semaphore
from time import sleep
#from traceback import print_exc
#from cStringIO import StringIO

from wx.lib import masked

from BitTornado.ConfigDir import ConfigDir
from BitTornado.bencode import bdecode
from BitTornado.download_btl import defaults as BTDefaults
from BitTornado.parseargs import parseargs
from BitTornado.zurllib import urlopen

from CH_BitTorrent_client.Actions.actions import makeActionList

if (sys.platform == 'win32'):
    from Utility.regchecker import RegChecker

from Utility.configreader import ConfigReader
from Utility.compat import convertINI, moveOldConfigFiles
from Utility.constants import * #IGNORE:W0611

#####
#
# Class: Utility
#
# Базовий «клейовий» клас, який містить зазвичай функції допомоги і допомагає
тримати трек об'єктів
#
#####
class Utility:
    def __init__(self, CH_BitTorrent_clientpath):
        self.CH_BitTorrent_clientpath = CH_BitTorrent_clientpath

        # Знаходимо директорію що зберігає конфігураційні файли.
        self.setupConfigPath()
        moveOldConfigFiles(self)

        self.setupConfig()

        # Встановлюємо файли мови
        self.lang = Lang(self)

        # Перетворюємо старі INI file
        convertINI(self)

        # створюємо торрент директорію (якщо потрібно)
        self.MakeBitTorrentDir()

        self.setupWebConfig()

        self.setupBitTorrentMakerConfig()

        self.setupBitTorrentList()

```

```

self.BitTorrents = { "all": [],
                    "active": {},
                    "inactive": {},
                    "pause": {},
                    "seeding": {},
                    "downloading": {} }

self.accessflag = Event()
self.accessflag.set()

self.invalidwinfilenamechar = ''
for i in range(32):
    self.invalidwinfilenamechar += chr(i)
self.invalidwinfilenamechar += '*/*/:<>?\\|'

self.FILESEM = Semaphore(1)

if (sys.platform == 'win32'):
    self.regchecker = RegChecker(self)

self.lastdir = { "save" : self.config.Read('defaultfolder'),
                "open" : "",
                "log": "" }

    self.CH_BitTorrent_clientquitting = False
# self.CH_BitTorrent_clientdonequitting = False

# Утримуємо трек останньої таблиці, яка розглядалася
self.lasttab = { "advanced" : 0,
                "preferences" : 0 }

self.languages = {}

# Записуємо треки в "ManagedList"
self.lists = {}

def setupConfigPath(self):
    configdir = ConfigDir()
    self.dir_root = configdir.dir_root

def getConfigPath(self):
    return self.dir_root

def setupConfig(self):
    defaults = {
        'defrentorwithdest': '1',
        'minport': str(random.randint(10000, 60000)),
        'maxport': '50000',
        'maxupload': '5',
        'maxuploadrate': '0',
        'maxdownloadrate': '0',
        'maxseeduploadrate': '0',
        'numsimdownload': '2',
        'uploadoption': '2',
        'uploadtimeh': '0',
        'uploadtimem': '30',
        'uploadratio': '100',
        'removeBitTorrent': '0',
        'setdefaultfolder': '0',
        'defaultfolder': 'c:\\',
        'defaultmovedir': 'c:\\',
        'mintray': '0',
        'trigwhenfinishseed': '1',
        'confirmonclose': '1',
        'kickban': '1',
        'notsameip': '1',
        'ipv6': '0',
        'ipv6_binds_v4': '1',
        'min_peers': '20',
    }

```

```

'max_initiate': '40',
'alloc_type': 'normal',
'alloc_rate': '2',
'max_files_open': '50',
'max_connections': '0',
'lock_files': '1',
'lock_while_reading': '0',
'double_check': '1',
'triple_check': '0',
'timeouttracker': '15',
'timeoutdownload': '30',
'timeoutupload': '1',
'scrape': '0',
'defaultpriority': '2',
'failbehavior': '0',
'language_file': 'english.lang',
'urm': '0',
'urmupthreshold': '10',
'urmdelay': '60',
'stripedlist': '0',
'mode': '1',
'window_width': '710',
'window_height': '400',
'detailwindow_width': '610',
'detailwindow_height': '500',
'prefwindow_width': '530',
'prefwindow_height': '400',
'prefwindow_split': '130',
'column4_rank': '0', # Заголовок
'column4_width': '150',
'column5_rank': '1', # Прогрес
'column5_width': '160',
'column6_rank': '2', # BT статус
'column6_width': '100',
'column7_rank': '8', # Пріоритет
'column7_width': '50',
'column8_rank': '5', # ETA
'column8_width': '85',
'column9_rank': '6', # Розмір
'column9_width': '75',
'column10_rank': '3', # DL швидкість
'column10_width': '65',
'column11_rank': '4', # UL швидкість
'column11_width': '60',
'column12_rank': '7', # %U/D розмір
'column12_width': '150',
'column13_rank': '9', # Повідомлення про помилку
'column13_width': '60',
'column14_rank': '-1', # # З'єднує сіди
'column14_width': '60',
'column15_rank': '-1', # # З'єднує піри
'column15_width': '60',
'column16_rank': '-1', # # Копіює
'column16_width': '60',
'column17_rank': '-1', # Прогрес по пірам
'column17_width': '60',
'column18_rank': '-1', # Розмір закачаних даних
'column18_width': '75',
'column19_rank': '-1', # Розмір даних, які ще треба закачати
'column19_width': '75',
'column20_rank': '-1', # Загальна швидкість
'column20_width': '80',
'column21_rank': '-1', # Найменування торренту
'column21_width': '150',
'column22_rank': '-1', # Місце призначення
'column22_width': '150',
'column23_rank': '-1', # Час для сидів
'column23_width': '85',
'column24_rank': '-1', # З'єднання

```

```

'column24_width': '60',
'column25_rank': '-1', # Опції відбору
'column25_width': '80',
'fastresume': '1',
'randomport': '1',
'savecolumnwidth': '1',
#   'forcenewdir': '1',
'upnp_nat_access': '0',
'buffer_write' : '4',
'buffer_read'  : '1',
'auto_flush'  : '0',
'associate'   : '1',
'movecompleted': '0',
'spew0_rank': '0', # Розрахунковий час завершення
'spew0_width': '24',
'spew1_rank': '1', # IP
'spew1_width': '132',
'spew2_rank': '2', # Локальний / Віддалений
'spew2_width': '24',
'spew3_rank': '3', # Середнє
'spew3_width': '72',
'spew4_rank': '4', #
'spew4_width': '24',
'spew5_rank': '5', #
'spew5_width': '24',
'spew6_rank': '6', # Середня швидкість завантаження
'spew6_width': '72',
'spew7_rank': '7', #
'spew7_width': '24',
'spew8_rank': '8', # Час завершення
'spew8_width': '24',
'spew9_rank': '9', #
'spew9_width': '24',
'spew10_rank': '10', # Закачано
'spew10_width': '84',
'spew11_rank': '11', # Віддано
'spew11_width': '84',
'spew12_rank': '12', # Прогрес пірів
'spew12_width': '72',
'spew13_rank': '-1', # Швидкість завантаження пірів
'spew13_width': '72',
'fileinfo0_rank': '0', # Ім'я файлу
'fileinfo0_width': '300',
'fileinfo1_rank': '1', # Розмір
'fileinfo1_width': '100',
'fileinfo2_rank': '2', # Прогрес
'fileinfo2_width': '60',
'fileinfo3_rank': '3', # MD5 Хеш
'fileinfo3_width': '200',
'fileinfo4_rank': '-1', # CRC32 Хеш
'fileinfo4_width': '200',
'fileinfo5_rank': '-1', # SHA1 Хеш
'fileinfo5_width': '200',
'fileinfo6_rank': '-1', # ED2K Хеш
'fileinfo6_width': '200',
'color_startup': '0000000000',
'color_disconnected': '100100100',
'color_noconnections': '2000000000',
'color_noincoming': '1501500000',
'color_nocomplete': '000000150',
'color_good': '0001500000',
'color_stripe': '245245245',
'display_interval': '0.8',
'listfont': '',
'diskfullthreshold': '0',
# 'showmenuicons': '1',
'icons_toolbarbottom': [ACTION_MOVEUP,
                        ACTION_MOVEDOWN,
                        ACTION_MOVEUP,

```

```

        ACTION_MOVEBOTTOM,
        -1,
        ACTION_CLEARCOMPLETED,
        -1,
        ACTION_PAUSEALL,
        ACTION_STOPALL,
        ACTION_UNSTOPALL,
        -1],
    'icons_toolbartop': [ACTION_ADDBITTORRENT,
        ACTION_ADDBITTORRENTNONDEFAULT,
        ACTION_ADDBITTORRENTURL,
        -1,
        ACTION_RESUME,
        ACTION_PAUSE,
        ACTION_STOP,
        ACTION_QUEUE,
        ACTION_REMOVE,
        -1,
        ACTION_SCRAPE,
        ACTION_DETAILS],
    'menu_listrightclick': [ACTION_RESUME,
        ACTION_STOP,
        ACTION_PAUSE,
        ACTION_QUEUE,
        ACTION_HASHCHECK,
        -1,
        ACTION_REMOVE,
        ACTION_REMOVEFILE,
        ACTION_EXPORTMENU,
        ACTION_CLEARMESSAGE,
        -1,
        ACTION_LOCALUPLOAD,
        ACTION_CHANGEPRIO,
        -1,
        ACTION_OPENFILEDEST,
        ACTION_OPENDEST,
        ACTION_CHANGEDEST,
        -1,
        ACTION_SCRAPE,
        ACTION_DETAILS],
#     'skipcheck': '0'
}

    configfilepath = os.path.join(self.getConfigPath(),
"CH_BitTorrent_client.conf")
    self.config = ConfigReader(configfilepath, "CH_BitTorrent_client",
defaults)
#     self.config = ConfigReader(configfilepath, "CH_BitTorrent_client")
#     self.config.defaults = defaults

    def setupWebConfig(self):
        defaults = {
            'webID': 'yourkeyword',
            'webIP': '127.0.0.1',
            'webport': '56667',
            'webautostart': '0',
            'allow_query': '1',
            'allow_delete': '1',
            'allow_clearcompleted': '1',
            'allow_add': '1',
            'allow_setparam': '0',
            'allow_getparam': '0',
            'allow_queue': '1',
            'allow_pause': '1',
            'allow_stop': '1',
            'allow_resume': '1',
            'allow_setprio': '1',
        }
}

```

```

        webconfigfilepath = os.path.join(self.getConfigPath(),
"webservice.conf")
        self.webconfig = ConfigReader(webconfigfilepath,
"CH_BitTorrent_client/Webservice", defaults)

    def setupBitTorrentMakerConfig(self):
        defaults = {
            'piece_size': '0',
            'comment': '',
            'created_by': '',
            'announcedefault': '',
            'announcehistory': '',
            'announce-list': '',
            'httpseeds': '',
            'makehash_md5': '0',
            'makehash_crc32': '0',
            'makehash_shal': '0',
            'startnow': '0',
            'saveBitTorrent': '2'
        }

        BitTorrentmakerconfigfilepath = os.path.join(self.getConfigPath(),
"maker.conf")
        self.makerconfig = ConfigReader(BitTorrentmakerconfigfilepath,
"CH_BitTorrent_client/BitTorrentMaker", defaults)

    def setupBitTorrentList(self):
        BitTorrentfilepath = os.path.join(self.getConfigPath(),
"BitTorrent.list")
        self.BitTorrentconfig = ConfigReader(BitTorrentfilepath, "list0")

    # Ініціалізація, яку доведеться зробити після того, як wx.App object
    # був створений
    def postAppInit(self):
        try:
            self.icon = wx.Icon(os.path.join(self.getPath(),
'icon_CH_BitTorrent_client.ico'), wx.BITMAP_TYPE_ICO)
        except:
            pass

        makeActionList(self)

    def getLastDir(self, operation = "save"):
        lastdir = self.lastdir[operation]

        if operation == "save":
            if not os.access(lastdir, os.F_OK):
                lastdir = self.config.Read('defaultfolder')

        if not os.access(lastdir, os.F_OK):
            lastdir = ""

        return lastdir

    def getPath(self):
        return self.CH_BitTorrent_clientpath

    def eta_value(self, n, truncate = 3):
        if n == -1:
            return '<unknown>'
        if not n:
            return ''
        n = int(n)
        week, r1 = divmod(n, 60 * 60 * 24 * 7)
        day, r2 = divmod(r1, 60 * 60 * 24)
        hour, r3 = divmod(r2, 60 * 60)
        minute, sec = divmod(r3, 60)

        if week > 1000:

```

```

        return '<unknown>'

    weekstr = '%d' % (week) + self.lang.get('l_week')
    daystr = '%d' % (day) + self.lang.get('l_day')
    hourstr = '%d' % (hour) + self.lang.get('l_hour')
    minutestr = '%02d' % (minute) + self.lang.get('l_minute')
    secstr = '%02d' % (sec) + self.lang.get('l_second')

    if week > 0:
        text = weekstr
        if truncate > 1:
            text += ":" + daystr
        if truncate > 2:
            text += "-" + hourstr
    elif day > 0:
        text = daystr
        if truncate > 1:
            text += "-" + hourstr
        if truncate > 2:
            text += ":" + minutestr
    elif hour > 0:
        text = hourstr
        if truncate > 1:
            text += ":" + minutestr
        if truncate > 2:
            text += ":" + secstr
    else:
        text = minutestr
        if truncate > 1:
            text += ":" + secstr

    return text

def getMetaInfo(self, src, openoptions = 'rb', style = "file"):
    if src is None:
        return None

    metaInfo = None
    try:
        metaInfo_file = None
        # Ми беремо url
        if style == "rawdata":
            return bdecode(src)
        elif style == "url":
            metaInfo_file = urlopen(src)
        # Ми беремо файл, який існує
        elif os.access(src, os.R_OK):
            metaInfo_file = open(src, openoptions)

        if metaInfo_file is not None:
            metaInfo = bdecode(metaInfo_file.read())
            metaInfo_file.close()
    except:
        if metaInfo_file is not None:
            try:
                metaInfo_file.close()
            except:
                pass
        metaInfo = None
    return metaInfo

def speed_format(self, s, truncate = 1, stopearly = None):
    return self.size_format(s, truncate, stopearly) + "/" +
self.lang.get('l_second')

def size_format(self, s, truncate = None, stopearly = None, applylabel =
True, rawsize = False, showbytes = False, labelonly = False, textonly = False):
    size = 0.0
    label = ""

```

```

if truncate is None:
    truncate = 2

if ((s < 1024) and showbytes and stopearly is None) or stopearly ==
"Byte":
    truncate = 0
    size = s
    text = "Byte"
elif ((s < 1048576) and stopearly is None) or stopearly == "KB":
    size = (s/1024.0)
    text = "KB"
elif ((s < 1073741824L) and stopearly is None) or stopearly == "MB":
    size = (s/1048576.0)
    text = "MB"
elif ((s < 1099511627776L) and stopearly is None) or stopearly ==
"GB":
    size = (s/1073741824.0)
    text = "GB"
else:
    size = (s/1099511627776.0)
    text = "TB"

if textonly:
    return text

label = self.lang.get(text)
if labelonly:
    return label

if rawsize:
    return size

# У цей момент, лише, приймає значення 0, 1, або 2
if truncate == 0:
    text = ('%.0f' % size)
elif truncate == 1:
    text = ('%.1f' % size)
else:
    text = ('%.2f' % size)

if applylabel:
    text += ' ' + label

return text

def makeNumCtrl(self, parent, value, integerWidth = 6, fractionWidth = 0,
min = 0, max = None, size = wx.DefaultSize):
    if size != wx.DefaultSize:
        autoSize = False
    else:
        autoSize = True
    return masked.NumCtrl(parent,
                           value = value,
                           size = size,
                           integerWidth = integerWidth,
                           fractionWidth = fractionWidth,
                           allowNegative = False,
                           min = min,
                           max = max,
                           groupDigits = False,
                           useFixedWidthFont = False,
                           autoSize = autoSize)

def MakeBitTorrentDir(self):
    BitTorrentpath = os.path.join(self.getConfigPath(), "BitTorrent")
    pathexists = os.access(BitTorrentpath, os.F_OK)
    # Якщо директорія торренту не існує, створюємо її
    if not pathexists:

```

```

os.mkdir(BitTorrentpath)

def RemoveEmptyDir(self, basedir, removesubdirs = True):
    # видаляємо піддиректорії
    if removesubdirs:
        for root, dirs, files in os.walk(basedir, topdown = False):
            for name in dirs:
                dirname = os.path.join(root, name)

                # Лише пробуємо видалити, якщо це існує
                if os.access(dirname, os.F_OK):
                    if not os.listdir(dirname):
                        os.rmdir(dirname)
    #видалення директорії
    if os.access(basedir, os.F_OK):
        if not os.listdir(basedir):
            os.rmdir(basedir)

def makeBitmap(self, bitmap, trans_color = wx.Colour(200, 200, 200)):
    button_bmp = wx.Bitmap(os.path.join(self.getPath(), 'icons', bitmap),
wx.BITMAP_TYPE_BMP)
    button_mask = wx.Mask(button_bmp, trans_color)
    button_bmp.SetMask(button_mask)
    return button_bmp

def makeBitmapButton(self, parent, bitmap, tooltip, event, trans_color =
wx.Colour(200, 200, 200)):
    tooltiptext = self.lang.get(tooltip)

    button_bmp = self.makeBitmap(bitmap, trans_color)

    ID_BUTTON = wx.NewId()
    button_btn = wx.BitmapButton(parent, ID_BUTTON, button_bmp, size =
wx.Size(button_bmp.GetWidth()+18, button_bmp.GetHeight()+4))
    button_btn.SetToolTipString(tooltiptext)
    parent.Bind(wx.EVT_BUTTON, event, button_btn)
    return button_btn

def getBTParams(self, skipcheck = False):
    # Створюємо BT параметри
    #####
    btparams = []

    btparams.append("--display_interval")
    btparams.append(self.config.Read('display_interval'))

    # Використовується тільки один порт
    btparams.append("--minport")
    btparams.append(self.config.Read('minport'))
    btparams.append("--maxport")
    btparams.append(self.config.Read('minport'))

#    btparams.append("--random_port")
#    btparams.append(self.config.Read('randomport'))

    #if self.config.Read('ipv6') == "1":
    #    btparams.append("--ipv6_enable")
    #    btparams.append(self.config.Read('ipv6'))
    #    btparams.append("--ipv6_binds_v4")
    #    btparams.append(self.config.Read('ipv6_binds_v4'))

    # Швидкий підсумок
    btparams.append("--selector_enabled")
    btparams.append(self.config.Read('fastresume'))

    btparams.append("--auto_kick")
    btparams.append(self.config.Read('kickban'))
    btparams.append("--security")
    btparams.append(self.config.Read('notsameip'))

```

```

btparams.append("--max_upload_rate")
btparams.append("0")

paramlist = [ "ip",
              "bind",
              "alloc_rate",
              "alloc_type",
              "double_check",
              "triple_check",
              "lock_while_reading",
              "lock_files",
              "min_peers",
              "max_files_open",
              "max_connections",
              "upnp_nat_access",
              "auto_flush" ]

for param in paramlist:
    value = self.config.Read(param)
    if value != "":
        btparams.append("--" + param)
        btparams.append(value)

config, args = parseargs(btparams, BTDefaults)

return config

# Перевіряємо, якщо рядок - дійсне Windows ім'я файлу (або ім'я програми,
якщо програма вірна)
# Якщо ім'я файлу недійсне: повертаємо виправлене ім'я
# Якщо ім'я файлу дійсне: повертає порожній рядок
def fixWindowsName(self, name, unit = False):
    if unit and (len(name) != 2 or name[1] != ':'):
        return 'c:'
    if not name or name == '.' or name == '..':
        return '_'
    if unit:
        name = name[0]
    fixed = False
    if len(name) > 250:
        name = name[:250]
        fixed = True
    fixedname = ''
    spaces = 0
    for c in name:
        if c in self.invalidwinfilenamechar:
            fixedname += '_'
            fixed = True
        else:
            fixedname += c
            if c == ' ':
                spaces += 1
    if fixed:
        return fixedname
    elif spaces == len(name):
        # contains only spaces
        return '_'
    else:
        return ''

def checkWinPath(self, parent, pathtocheck):
    if pathtocheck and pathtocheck[-1] == '\\\' and pathtocheck !=
'\\\\':
        pathitems = pathtocheck[:-1].split('\\\\')
    else:
        pathitems = pathtocheck.split('\\\\')
    nexttotest = 1
    if self.isPathRelative(pathtocheck):
        # Відносний шлях до файлу

```

```

# Порожній відносний шлях до файлу дозволяється
if pathtocheck == '':
    return True
fixedname = self.fixWindowsName(pathitems[0])
if fixedname:
    dlg = wx.MessageDialog(parent,
                           pathitems[0] + '\n' + \
                           self.lang.get('invalidwinname') + '\n'+ \
                           self.lang.get('suggestedname') + '\n\n' + \
                           fixedname,
                           self.lang.get('error'), wx.ICON_ERROR)

    dlg.ShowModal()
    dlg.Destroy()
    return False
else:
    # Абсолютний шлях
    # Абсолютний шлях повинен мати лише одне '\'
    if not '\\' in pathtocheck:
        dlg = wx.MessageDialog(parent, pathitems[0] + '\n' +
self.lang.get('errorinvalidpath'),
                               self.lang.get('error'), wx.ICON_ERROR)

        dlg.ShowModal()
        dlg.Destroy()
        return False
    if pathtocheck[:2] != '\\\\':
        # Not a мережний шлях
        fixedname = self.fixWindowsName(pathitems[0], unit = True)
        if fixedname:
            dlg = wx.MessageDialog(parent,
                                   pathitems[0] + '\n' + \
                                   self.lang.get('invalidwinname') + \
                                   fixedname,
                                   self.lang.get('error'),
wx.ICON_ERROR)

            dlg.ShowModal()
            dlg.Destroy()
            return False
        else:
            # Мережний шлях
            nexttotest = 2

            for name in pathitems[nexttotest:]:
                fixedname = self.fixWindowsName(name)
                if fixedname:
                    dlg = wx.MessageDialog(parent, name + '\n' +
self.lang.get('errorinvalidwinname') + fixedname,
                                            self.lang.get('error'), wx.ICON_ERROR)

                    dlg.ShowModal()
                    dlg.Destroy()
                    return False

            return True

def isPathRelative(self, path):
    if len(path) < 2 or path[1] != ':' and path[:2] != '\\\\':
        return True
    return False

# Отримуємо словник з інформацією про шрифт
def getInfoFromFont(self, font):
    default = wx.SystemSettings.GetFont(wx.SYS_DEFAULT_GUI_FONT)

    try:
        if font.Ok():
            font_to_use = font
        else:
            font_to_use = default

```

```

fontname = font_to_use.GetFaceName()
fontsize = font_to_use.GetPointSize()
fontstyle = font_to_use.GetStyle()
fontweight = font_to_use.GetWeight()

fontinfo = {'name': fontname,
            'size': fontsize,
            'style': fontstyle,
            'weight': fontweight }
except:
    fontinfo = {'name': "",
                'size': 8,
                'style': wx.FONTSTYLE_NORMAL,
                'weight': wx.FONTWEIGHT_NORMAL }

return fontinfo

def getFontFromInfo(self, fontinfo):
    size = fontinfo['size']
    name = fontinfo['name']
    style = fontinfo['style']
    weight = fontinfo['weight']

    try:
        font = wx.Font(size, wx.DEFAULT, style, weight, faceName = name)
    except:
        font = wx.SystemSettings.GetFont(wx.SYS_DEFAULT_GUI_FONT)

    return font

# Робимо вхід для меню, що вискакує
def makePopup(self, menu, event = None, label = "", extralabel = "", bindto
= None):
    text = ""
    if label != "":
        text = self.lang.get(label)
    text += extralabel

    newid = wx.NewId()
    if event is not None:
        if bindto is None:
            bindto = menu
        bindto.Bind(wx.EVT_MENU, event, id = newid)
    menu.Append(newid, text)
    return newid

```