

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 завантаження
файлів з мережі за технологією торренту”

Виконав здобувач вищої освіти
II курсу, групи КН-22М-1
ОПП «Комп’ютерні науки»
спеціальності 122 «Комп’ютерні науки»
_____ Заєць А.О.
« ____ » _____ 2023 р.

Керівник проекту
кандидат технічних наук, доцент
_____ Доренський О.П.
« ____ » _____ 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 122 "Комп'ютерні науки"
Освітньо-професійна (освітньо-наукова) програма "Комп'ютерні науки"

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 6 » вересня 2023 року

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

Зайцю Артему Олександровичу

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та програмна реалізація системи завантаження файлів з мережі за технологією торренту

2. Керівник роботи Доренський Олександр Павлович, канд. техн. наук, доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 32-13 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 завантаження файлів з мережі за технологією торренту

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

Заєць А.О. Дослідження та програмна реалізація системи завантаження файлів з мережі за технологією торренту. 122 Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2023.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи завантаження файлів з мережі за технологією торренту.

Метою розробки є дослідження та програмна реалізація системи завантаження файлів з мережі за технологією торренту.

Об'єктом дослідження є процес завантаження файлів з мережі за технологією торренту.

Предметом дослідження є методи завантаження файлів з мережі за технологією торренту.

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

Результат роботи – програмна реалізація системи завантаження файлів з мережі за технологією торренту.

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'ютерні науки, торрент

ABSTRACT

Zaiets A.O. Research and software implementation of a system for downloading files from the network using torrent technology. 122 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2023.

In this graduation thesis for the second (master's) level of higher education, software is developed, which is intended for the system of downloading files from the network using torrent technology.

The goal of the development is the research and software implementation of a system for downloading files from the network using torrent technology.

The object of the study is the process of downloading files from the network using torrent technology.

The subject of the study is methods of downloading files from the network using torrent technology.

The research methods are based on the methods of the theory of building computer networks, the methods of mathematical statistics, and the methods of software development.

The result of the work is the software implementation of the system for downloading files from the network using torrent technology.

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

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

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

The program was developed in the Python environment.

Keywords: computer science, torrent

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	6
1.1 Призначення системи.....	6
1.2 Область застосування.....	6
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	9
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	9
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	15
2.3 Розгорнута постановка завдання	17
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	19
3.1 Опис функціонування системи	19
3.2 Розробка структурної схеми.....	26
3.3 Розробка функціональної схеми	34
3.4 Розробка діаграми процесів.....	41
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	43
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	43
4.2 Захист розробленого програмного забезпечення.....	56
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	63
6 НАУКОВА НОВИЗНА	65

						ВКРМ-122.23.0008.00.00.ПЗ		
Вим	Арк.	№ докум.	Підп.	Дата				
Розроб.	Засць А.О.				Дослідження та програмна реалізація системи завантаження файлів з мережі за технологією торренту	Літ.	Аркуш	Аркушів
Перев.	Доренський О.П.					М	1	105
Н.контр.	Коваленко А.С.					ЦНТУ КН-22М-1		
Затв.	Смірнов О.А.							

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-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

BitTorrent – це законний протокол передачі файлів, і його використання, що називається торрент-завантаженням, є законним, якщо вміст можна завантажувати або завантажувати на законних підставах. Однак використання його для завантаження матеріалів, захищених авторським правом – наприклад, абсолютно нового фільму – без дозволу власника авторських прав є незаконним. З 2010 по 2011 рік у Сполучених Штатах подали до суду понад 200 000 користувачів за завантаження матеріалів, захищених авторським правом, за допомогою BitTorrent.

З того часу судові позови замінили авторські тролі, які уповноважені діяти від імені власників авторських прав.

Оскільки технологія BitTorrent розкриває IP-адреси пристроїв, підключених до її потоку даних, який називається торрентом, тролі можуть визначити, з яких адрес завантажуються файли.

Тролі надсилають розрахункові листи постачальникам послуг Інтернету (ISP), які потім пересилають листи своїм клієнтам.

Типовий запит на врегулювання у 2017 році від групи, відомої як Rights Enforcement, становив 300 доларів США. Оскільки IP-адреси не вважаються юридичними особами, тролі рідко мають імена фізичних осіб і тому не можуть продовжувати збір.

1.2 Область застосування

Завантаження патчів, модифікацій і оновлень для старіших, непідтримуваних відеоігор і створених користувачами модифікацій для підтримуваних ігор є вагомою причиною для доступу до BitTorrent.

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

Доступ до класичних фільмів, фільмів В, інді-музики, книг і програмного забезпечення у відкритому доступі або через ліцензії Creative Commons є ще одним законним випадком використання BitTorrent. Прикладами легальних сайтів є, зокрема, Public Domain Torrents та інді-сайт Vodo.

До завантаження торрентів слід підходити обережно. BitTorrent має репутацію технології, яка використовується для піратства фільмів, ігор та іншого вмісту, захищеного авторським правом. Інтернет-провайдери це знають і часто надсилають користувачам BitTorrent листи з попередженнями та навчальні матеріали щодо боротьби з піратством.

У повторних порушників пропускна спроможність може бути зменшена або облікові записи призупинено. Інтернет-провайдери можуть не відрізнити законні сайти BitTorrent, що розповсюджують матеріали з ліцензіями Creative Commons, від тих, які обмінюються піратським вмістом.

Крім того, хакери часто використовують безкоштовний вміст як приманку зі зловмисним програмним забезпеченням, щоб завдати шкоди комп'ютерам користувачів або здійснити атаки. Щоб мінімізувати ризик, з'явилися потокові сайти BitTorrent, які стверджують, що перевіряють файли на наявність вірусів та інших шкідливих програм.

Ще одна небезпека – видимість. Адреси ISP користувачів і вміст, який вони передають, бачать усі користувачі BitTorrent – і ця інформація часто відстежується власниками авторських прав, правоохоронними органами, рекламодавцями та хакерами. Користувачам торрент-файлів важливо підтримувати повну безпеку в Інтернеті, використовуючи програмне забезпечення для безпеки в Інтернеті, оновлюючи визначення зловмисного програмного забезпечення та встановлюючи виправлення операційної системи, щойно вони стануть доступними. Шифрування підключення до Інтернету за допомогою віртуальної приватної мережі (VPN) для маскуванню IP-адреси також допомагає, але це не надійно.

Як зазначалося вище, переконайтеся, що у вас є надійний інструмент безпеки в Інтернеті та підключіться до VPN, щоб замаскувати свою IP-адресу. Потім завантажте та налаштуйте клієнт BitTorrent. Щоб знайти вміст,

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

скористайтеся веб-браузером, щоб відвідати популярні сайти трекерів BitTorrent. Використовуйте клієнт BitTorrent, щоб фактично завантажити потрібні файли.

Пам'ятайте, хоча сам BitTorrent є легальним, багато вмісту, доступного через інших користувачів, є піратським. Завантажуйте лише легальний вміст і завжди перевіряйте всі файли на наявність шкідливих програм.

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

КБГПЗ - 2023

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

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

У даному розділі проведемо огляд торрент-клієнтів.

µТоррент

µТоррент (також відомий як uТоррент або microТоррент) – безкоштовний BitТоррент-клієнт для Microsoft Windows, написаний на С++ і який має невеликий розмір при досить великій функціональності.

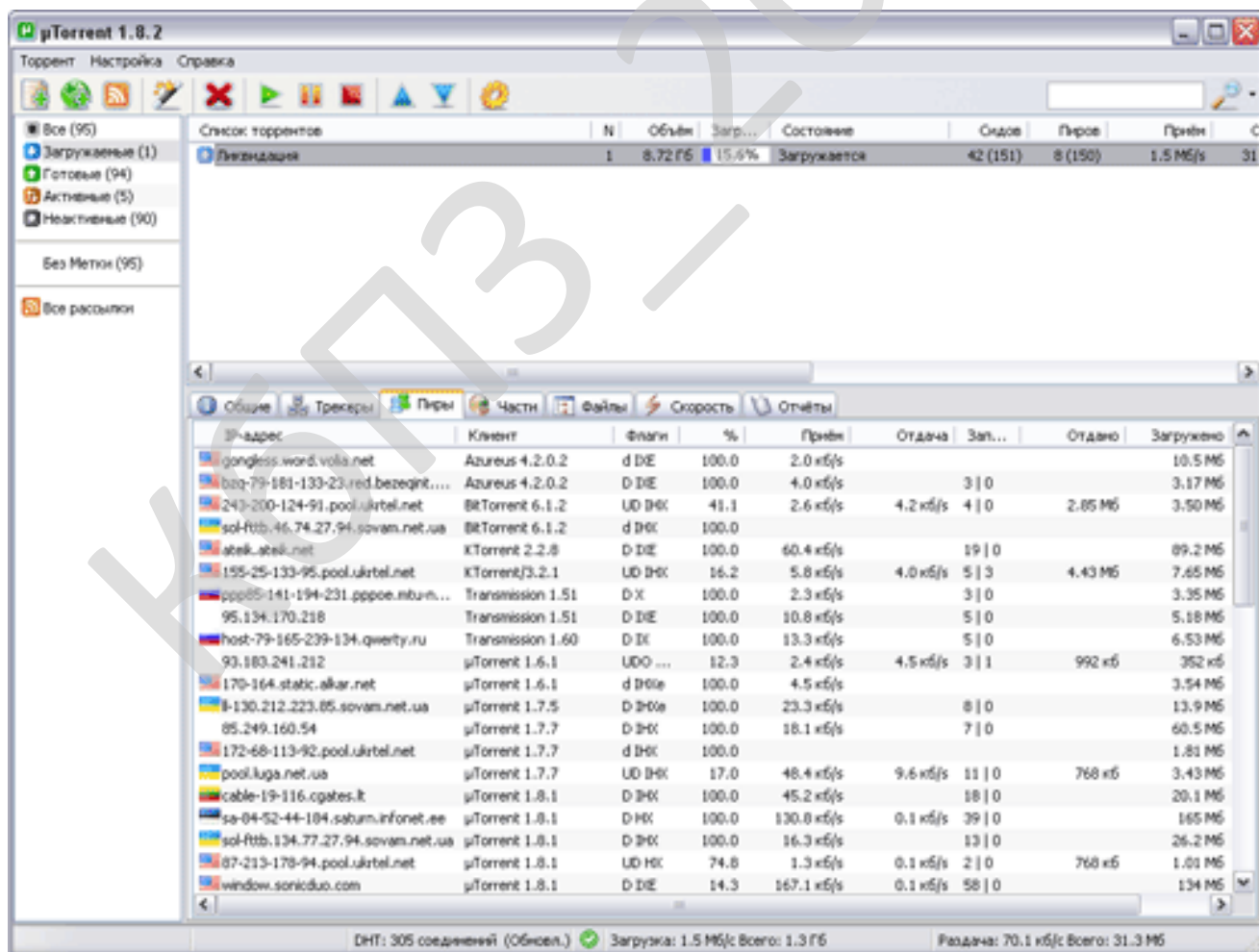


Рисунок 2.1 – Інтерфейс користувача µТоррент

Підтримка ОС: Microsoft Windows.

Особливості:

- обмеження максимальних швидкостей завантаження й віддачі;
- налаштування цих обмежень залежно від часу;
- обмеження максимальних швидкостей кожного завдання;
- налаштування кешування файлів на жорсткому диску;
- підтримка DHT і Peer Exchange;
- режим початкової роздачі (суперсід);
- автоматичне закачування торрентів зі стрічки новин RSS;
- підтримка проксі-серверів;
- можливість підключення до трекеру по HTTPS;
- шифрування протоколу, сумісне з Azureus, Mainline і BitComet;
- підтримка Юнікоду у всіх версіях Windows;
- підтримка UPnP у всіх версіях Windows;
- повна підтримка операційної системи Windows Vista;
- налаштування інтерфейсу програми;
- локалізація на 38 мов, включаючи російську;
- веб-інтерфейс (віддалене керування програмою із браузера) – доступний у бета-версії;
- можливість запуску на GNU/Linux і Macintosh за допомогою Wine.

BitТоррент (Mainline)

BitТоррент або Mainline – кроссплатформене програмне забезпечення для файлообміну по протоколу BitТоррент Написаний мовою Python. Існують версії для ОС Microsoft Windows, GNU/Linux, Mac OS. Можливий пошук файлів по мережі (у вікні браузера). Після придбання μТоррент Bitторрент для Windows, починаючи з 6-й версії, ґрунтується на його вихідному коді й має практично однаковий з ним інтерфейс, а розробка версій для GNU/Linux і Mac OS була заморожена.

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

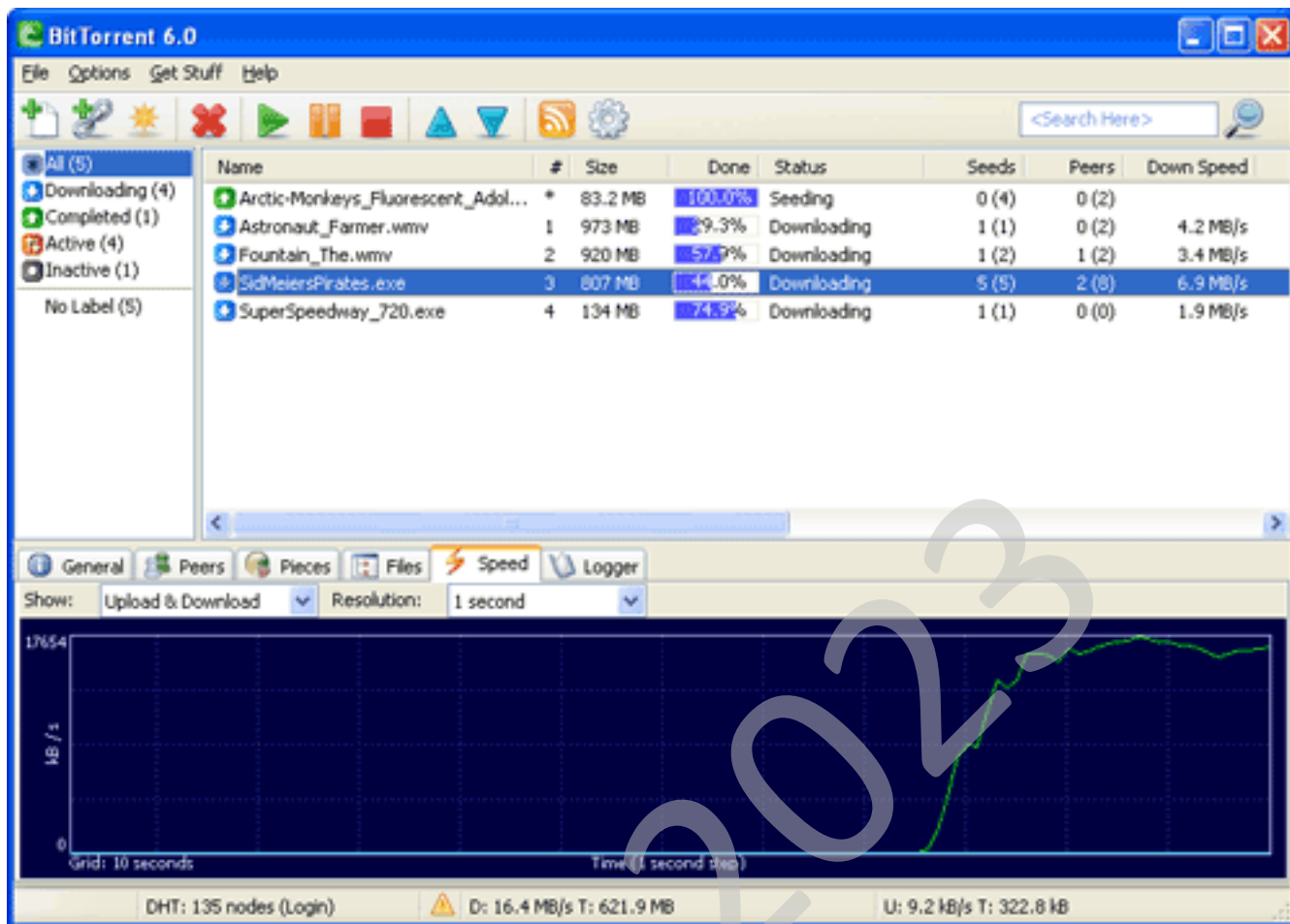


Рисунок 2.2 – Інтерфейс користувача ВіТторрент

Підтримка ОС: Microsoft Windows, GNU/Linux і Mac OS

Особливості по суті, ідентичні особливостям μТоррент.

Deluge

Deluge – це ВіТторрент-клієнт, створений на Python і GTK+. Deluge у цей час доступний для POSIX-сумісних операційних систем. Deluge створений бути нативним, повністю відповідному оточенню GTK, таким як GNOME і Xfce. Існує офіційний порт під Windows. Програма використовує бібліотеку libtorrent.

- UPn і NAT-PMP.
- Підтримка проксі для пірів, трекерів, DHT і Web-Seed.
- Частні торренти.
- Ручне додавання пірів.

Deluge має систему плагінів, деякі з них включені в Deluge:

- Імпорт «чорних» списків.
- Desired Ratio.
- Додаткова статистика.
- Locations.
- Network Activity Graph.
- Network Health Monitor.
- RSS Broadcatcher.
- Створення торрентів.
- Торрент Notification.
- Пошук торрентів.

Vuze (Azureus)

Вільне кроссплатформене програмне забезпечення для роботи з файлообмінними мережами за протоколом BitTorrent з підтримкою анонімного обміну даними по протоколах I2P і Tor. Написаний мовою Java. Функції графічної оболонки виконує бібліотека SWT. Vuze поширюється під ліцензією GPL. Локалізація 36 мов, включаючи російську, а також має докладну он-лайн довідку на англійській, французькій, німецькій і польській мовах.

Підтримка ОС: Microsoft Windows і GNU/Linux.

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

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 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, їхня 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.

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

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: 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. КORTEЖ 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.

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

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 завантаження файлів з мережі за технологією торренту.

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 методикку 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

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

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.

КБПЗ-2023

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Насправді, торренти – це онлайн-файли, які передаються через децентралізовану однорангову мережу, також звану мережею P2P, з BitTorrent протокол. Ці торрент-файли не зберігаються в централізованому місці, а зберігаються окремо в однорангових комп'ютерах, які приєдналися до мережі P2P.

Але який зв'язок між торрентом і торрент-клієнтом? Коли ви хочете завантажити файли з торрент-файлів, торрент-клієнт може проаналізувати детальну інформацію, включаючи ім'я, розмір, розташування та інше, що міститься у файлах торрент-файлів, а потім зв'язатися з кожним завантажувачем, щоб завантажити ці файли на ваш комп'ютер.

Однорангові служби обміну файлами, такі як торренти, часто блокуються провайдерами (Інтернет-провайдери) та мережевих адміністраторів, особливо коли ви перебуваєте в обмежених місцях, таких як офіси, коледжі та школи, головним чином через те, що торрент-файли споживають велику пропускну здатність. Більш того, іноді торрент-клієнти можуть викликати проблеми з вашим комп'ютером через неминучу рекламу та надзвичайно повільну швидкість завантаження.

Якщо ви зіткнулися зі складними ситуаціями, перерахованими вище, або ви просто не хочете нічого встановлювати на свій комп'ютер, а також завантажувати торрент-файли на високій швидкості за допомогою простих кроків, на щастя, є 3 найкращі надійні способи завантаження торрент-файлів онлайн без клієнта за допомогою за допомогою MultCloud, Bitport і Seedr.

Оскільки ви хочете безпосередньо завантажувати торрент без клієнта на свій комп'ютер, тут ми пропонуємо 3 чудові інструменти, які допоможуть вам

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

виконати завдання завантаження без встановлення будь-якого торрент-клієнта на ваш комп'ютер. Серед цих корисних інструментів MultCloud і Bitport постачаються на зручному веб-сайті з різними додатковими функціями, а Seedr дозволяє використовувати через розширення Chrome.

Рішення 1: Як завантажувати торренти без клієнта за допомогою MultCloud

MultCloud є безкоштовним хмарний файловий менеджер який дозволяє об'єднати всі ваші хмарні диски та разом керувати файлами, що зберігаються в цих хмарах, в одному інтерфейсі. Ви можете не тільки передавати, синхронізувати або створювати резервні копії даних із хмари в хмару в MultCloud, але й керувати своїми хмарними обліковими записами за допомогою функцій завантаження, завантаження, копіювання, перейменування, перегляду, вирізання. Більше того, ви можете ділитися файлами з різних хмар одержувачам одразу в MultCloud за допомогою режимів Public Share, Private Share та Source Share.

Таким чином, ви можете використовувати одну з відмінних функцій MultCloud під назвою **Remote Upload**, щоб завантажувати торрент без клієнта безпосередньо на ваші хмарні диски. Таким чином, вас не турбуватиме повільна швидкість передачі під час використання торрент-клієнта, оскільки завдання завантаження швидко виконується фоновією програмою MultCloud, але не вашою мережею.

Крім того, коли ви використовуєте Remote Upload для завантаження торрент-файлів у свої хмари, ви можете отримати доступ до цих файлів будь-де через будь-який пристрій, який може увійти у ваші хмари, не займаючи пам'яті ваших пристроїв. Ви також можете завантажити їх на свій комп'ютер або мобільний телефон із хмарних дисків із більшою швидкістю завантаження.

Тут ми беремо рішення приблизно завантажити торрент на Google Drive як приклад:

Крок 1: Зареєструватися MultCloud безкоштовно.

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

Крок 2. Натисніть «Додати хмару» та дотримуйтесь інструкцій у спливаючому вікні, щоб додати Google Drive та інші хмари до MultCloud.

MultCloud підтримує понад 30 провідних хмарних служб, включаючи Google Drive, MEGA, Dropbox, OneDrive, Flickr тощо. Отже, ви можете вільно додавати скільки завгодно хмар у MultCloud.

Крок 3. Увійдіть у свій обліковий запис Google Drive у MultCloud і виберіть «Віддалене завантаження» у спадному меню.

Використовуйте віддалене завантаження, щоб завантажити торрент без клієнта

Крок 4: Натисніть «Додати торрент» у спливаючому вікні. Потім знайдіть торрент-файл в іншому спливаючому вікні та натисніть «Відкрити».

Додайте торрент на свій Cloud Drive

Крок 5: Виберіть цільові файли, які ви хочете завантажити з торрента. Потім натисніть фіолетову кнопку «Завантажити». І завдання завантаження виконано.

Поради:

– Як безкоштовний користувач MultCloud ви можете створити одне завдання віддаленого завантаження з одним торрентом або іншим видом посилання для виконання одночасно. І якщо ви оновлюєте ваш обліковий запис MultCloud на вищій рівень, ви можете створити щонайбільше 5 завдань для одночасного виконання.

– Навіть MultCloud не встановлює обмежень на швидкість передачі даних, що означає, що ви можете використовувати Remote Upload для завантаження торрент-файлів із найвищою швидкістю.

Рішення 2: завантажуйте торренти без клієнта за допомогою Bitport

Bitport є одним із найпопулярніших хмарних торрент-сервісів, який дозволяє завантажувати торренти через його хмару, а потім на ваші пристрої з високою швидкістю. Якщо ваші торрент-файли досить базові, ви можете спробувати цей інструмент для завантаження торрент-файлів за допомогою веб-браузера.

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

Крок 1. Створіть обліковий запис Bitport або скористайтесь обліковим записом Facebook, щобувійти.

Крок 2. Знайдіть торрент-файл на своєму комп'ютері або просто перетягніть торрент-файл у Bitport.

Крок 3: Натисніть «Додати новий торрент». Потім ви можете знайти всі файли, завантажені з торрента, у списку «Готово».

Крок 4. Ви можете вибрати деякі файли для завантаження на свій комп'ютер або просто завантажити папку як файл zip.

Завантажте торрент-файл із Bitport

Рішення 3: завантажуйте торрент-файли онлайн без клієнта за допомогою Seedr

Іншим чудовим інструментом, який допоможе вам завантажити торрент без торрент-клієнта, є Seedr, який не обмежує швидкість завантаження торрент-файлів. Обмеження файлу становить 2 ГБ, що є як загальною ємністю пам'яті, так і максимальним розміром одного завдання завантаження торрент-файлу. Ви можете передавати лише один торрент за раз через Seedr, але можете завантажувати кілька готових торрентів одночасно.

Більше того, якщо ви використовуєте Google Chrome як щоденний браузер, ви можете швидко завантажити торрент через розширення Chrome Bitport, клацнувши торрент правою кнопкою миші та вибравши «Додати до Seedr».

Крок 1: ЗавантажтеSeedrРозширення Chrome із браузера Chrome.

Крок 2: зареєструйтесь у Seedr за допомогою електронної пошти.

Крок 3. Натисніть кнопку завантаження та знайдіть торрент-файл на своєму комп'ютері. Потім натисніть «Відкрити» у спливаючому вікні.

Крок 4. Ви можете двічі клацнути завантажену папку, щоб дізнатися цільові файли, або натиснути кнопку завантаження поруч із папкою, щоб завантажити файли на свій комп'ютер.

Завантажте торрент онлайн без клієнта в Seedr

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

Переглянувши 3 популярних способи завантаження торрентів без клієнта, ви, можливо, вже дізналися, наскільки просто завантажувати торренти онлайн за допомогою MultCloud, Bitport або Seedr.

Порівняно з Bitport і Seedr, ви також можете виявити, що MultCloud може запропонувати вам більше можливостей, особливо коли ви працюєте з хмарними дисками. Насправді, ви можете перенести всі файли Google Drive з одного облікового запису в інший або відразу в інші ваші хмари, щоб упорядкувати файли, що зберігаються у ваших хмарах, і зберегти сховище ваших хмар, одночасно досягаючи чудового керування хмарними дисками.

BitTorrent Web – це простий у користуванні онлайн торрент-клієнт, який використовує ваш браузер за умовчанням. Якщо ви вже знаєте, де знайти свій торрент-файл, ви можете клацнути його або перетягнути торрент у вікно браузера BitTorrent Web. Крім того, ви можете скористатися полем пошуку у верхній частині веб-вікна BitTorrent, щоб знайти завантажений торрент-файл. Якщо у вас встановлено безпечний торрент-сканер, який пропонується вам невдовзі після встановлення BitTorrent Web, він допоможе вам знаходити та завантажувати торрент-файли з результатів пошуку, тобто вам не потрібно відвідувати потенційно шкідливі веб-сайти. Ви можете вказати конкретний жорсткий диск або папку, куди потрібно завантажувати торренти. Ви також можете змінити папку завантаження за замовчуванням у налаштуваннях BitTorrent Web. Якщо файл є відео- чи аудіофайлом, ви можете передавати торрент-файли в медіапрогравачі, вбудованому у веб-торрент-клієнт вашого браузера.

Як транслювати торрент файли

BitTorrent Web спрощує потокову передачу торрент-файлів. Після встановлення у ваш улюблений браузер веб-торрент-клієнт дозволяє завантажувати будь-який тип файлу у вибрану вами папку. Однак, якщо файл є аудіо чи відео, ви можете передавати його у вікні браузера за допомогою вбудованого програвача. Найкраще те, що вам не потрібно чекати, поки файл

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

закінчиться завантаження - ви можете відтворити його майже миттєво після початку завантаження. Коли ви створите свою бібліотеку завантажених файлів, ви можете використовувати BitTorrent Web як улюблений медіаплеер для потокової передачі онлайн-торрентів.

Швидкість BitTorrent – це функція, яка розширює традиційний досвід наших торрент-клієнтів на базі Windows і Mac. Під час завантаження або оновлення до останньої версії µTorrent Classic для Windows, або BitTorrent або µTorrent Web, цифровий криптовалютний гаманець і баланс токенів ВТТ буде автоматично ввімкнено.

Під час завантаження торрентів BitTorrent Speed автоматично призначає BitTorrent (ВТТ) іншим користувачам для більшої швидкості. Просто використовуйте торрент-програму, як зазвичай, більше нічого не потрібно вивчати чи робити.

Коли завершується завантаження торрент-файлу, ваш клієнт BitTorrent може автоматично заробляти ВТТ шляхом заповнення. Чим довше ви роздаєте торренти, тим більше торрент-токенів ВТТ ви зможете заробити.

Інформаційна панель

Відстежуйте та керуйте своїми прибутками та покращеною швидкістю завантаження в одному місці.

Гаманець

Контролюйте свої активи ВТТ у зашифрованому гаманці. Надсилайте або отримуйте токени простим клацанням.

Переваги

Причини спробувати швидкість BitTorrent.

Простий у використанні

Переваги швидкості BitTorrent автоматизовані, включаючи ставки токенів для швидшого завантаження та винагороди токенів ВТТ. Вам не потрібно вчитися нічого нового.

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

завантаження. Якщо вам потрібно оптимізувати швидкість завантаження BitTorrent для одного торрента, майте це на увазі, перш ніж розпочати кілька завантажень. Або, якщо торренти вже завантажуються, призупиніть кілька, щоб пришвидшити пріоритетне завантаження.

Коли завершиться завантаження файлу, дайте йому почати роботу, щоб отримати BitTorrent (BTT)

Легко знайти торренти

Використовуйте утиліту безпечного сканування торрент-файлів, щоб швидко знаходити, завантажувати та відтворювати торренти за кілька простих кроків.

Відтворюйте торрент-файли відразу

Відтворюйте торрент-файли та магнітні посилання під час їх завантаження у вікні браузера.

Простий, інтуїтивно зрозумілий інтерфейс

Веб-торрент-клієнт встановлюється у ваш улюблений веб-переглядач і спрощує завантаження торрент-файлів.

Завантажте будь-який тип файлу

BitTorrent Web призначений не лише для відео. Завантажте будь-який файл для локального зберігання у вибраній вами папці.

Економте мережеві ресурси

Встановіть обмеження швидкості завантаження/завантаження, щоб звільнити ресурси мережі для виконання інших завдань в Інтернеті.

3.2 Розробка структурної схеми

BitTorrent-трекер – веб-сервер, що здійснює координацію клієнтів BitTorrent. Основна функція BitTorrent-трекеру – обробка запитів клієнтів. Кожний клієнт періодично звертається до трекеру із запитом, у якому зазначені:

– SHA-1 словника info (інфогеш);

- порт, на якому клієнт чекає з'єднань від інших клієнтів;
- кількість даних, якими клієнт встиг обмінятися з іншими клієнтами;
- і деяка інша інформація.

Клієнтський вузол відправляє трекеру HTTP-запит типу GET. Відповіддю служить Bencode-словник з інформацією про раніше звернулися до трекеру вузлах, що також беруть участь у роздачі. Дана інформація використовується вузлами для встановлення прямих зв'язків один з одним.

Якщо в BitTorrent-клієнті відкрито декілька активних torrent-файлів, то для кожного використовується окремий запит, ніяк не пов'язаний з іншими. Інтервали між запитами по кожному окремо взятому торренту залежать від налаштувань клієнта й звичайно становлять від 30 до 60 хвилин.

У мережі BitTorrent файл розбивається на частини, а потім розповсюджується по мережі. Фрагмент є одиницею передачі, і коли одноранговий вузол має всі фрагменти, він об'єднує їх і повторно створює весь файл.

Важливість стратегії вибору фрагмента

Наявність стратегії вибору частин є важливою, тому що, якщо кожен окремий пір починає першим із першої частини, це збільшить навантаження на однолітків/сідів, які їх мають; тим самим зменшуючи швидкість розподілу.

Що, якщо перед тим, як хтось зможе завантажити останні кілька фрагментів, розсівач (з усіма фрагментами) залишить мережу? жоден із розпусників не зможе завершити завантаження.

Вибір найрідкісніших перших творів

Основна ідея: віддавати пріоритет завантаженню найрідкіснішого твору в мережі.

Переваги

Поширення насіння

Стратегія вибору першої рідкісної частини гарантує, що лише «нові» частини (які не містять інших п'явок) завантажуються з розсівача. Отримуємо швидкий розподіл шматків у мережі та знижене навантаження на сівалку.

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

Збільшує швидкість завантаження

Оскільки більше однолітків мають різні частини, вони можуть швидко торгувати між собою та швидше завершувати завантаження.

Увімкнення завантаження

Коли у одного з колег є рідкісний твір, кожен інший буде зацікавлений у тому, щоб завантажити його. Через взаємно-поступальний рух він часто відключається від інших, отримуючи кращу швидкість завантаження.

Запобігання рідкісним відсутнім фрагментам

Встановлюючи пріоритет для завантаження найрідкісніших предметів, ми гарантуємо, що рідкісні предмети не зникнуть з мережі, навіть коли сідер залишиться.

Як обчислити найрідкісніший шматок?

Є два способи дізнатися, які фігури є у однолітків

1. Мати повідомлення: партнер у мережі транслює фрагменти, які має.
2. Повідомлення Bitfield: під час початкового рукостискання партнер надсилає повідомлення Bitfield, яке містить фрагменти, які він містить.

Кожен одноранговий вузол підтримує доступність одиниць у своєму наборі однорангових пристроїв і використовує її для обчислення найрідкіснішої частини.

Випадкова перша політика

Коли партнер приєднується до мережі, щоб активно брати участь, йому потрібно отримати першу частину якнайшвидше, і, отже, замість того, щоб спочатку вибирати найрідкіснішу, вона шукає випадкові частини.

Суворі політика пріоритетів

Файл розбивається на частини, а частина розбивається на блоки. Блоки – це те, що передається. Отже, коли витягується блок фрагмента, ми надаємо пріоритет завантаженню всіх блоків того самого фрагмента, перш ніж переходити до нового.

частина даних, при цьому багато користувачів після одержання потрібного файлу відразу закривають торрент-клієнт. У результаті доступність торренту знижується, і на первісних розповсюджувачів файлу лягає невинувато більше навантаження.

Як правило, приватні трекери записують статистику (рейтинг) кожного користувача по кількості «завантаженого» і «відданого» іншим користувачам, і вимагають підтримки деякого мінімального співвідношення цих двох величин. Як результат, доступність і швидкість завантаження торрентів на приватному трекері звичайно вище, ніж на відкритому.

Для ідентифікації конкретного клієнта трекер використовує або IP-адресу користувача, або унікальний для кожного користувача ключ (так званий user key або pass key), що додається трекером у torrent-файл при завантаженні його користувачем.

У рейтингової системи є й негативна сторона: дискримінація користувачів, що мають вузький канал віддачі. Часто в такого користувача віддача зовсім не йде; причина тому – інші користувачі, що мають більше широкий upload-канал і віддають значно більше й швидше. Такому користувачеві дуже складно відновлювати свій рейтинг, особливо якщо він має справу з малозатребуваними файлами. У підсумку перебування на приватному трекері обертається для нього постійним головним болем про способи підтримки рейтингу.

Для підтримки закритих роздач у torrent-файлах є прапор private, установка якого приводить до того, що клієнти, що правильно підтримують цей прапор, не роздають цей торрент в обхід трекеру (наприклад, через DHT). Тому що більшість torrent-файлів створюється без цього прапора, те раздаючий після завантаження свіжествореного файлу на трекер повинен заново скачати його, тому що трекер виставляє цей прапор, змінюючи торрент.

Реалізації трекерів

Існують різні реалізації трекерів: у вигляді окремого веб-сервера, у вигляді модуля для стороннього HTTP-сервера (наприклад, Apache) або у вигляді доповнення до движка сайту, написаного, приміром, на PHP або JSP.

Каталог BitTorrent

Каталог BitTorrent – список torrent-файлів, призначений для їхнього пошуку й завантаження через BitTorrent. Звичайно веб-сайт або IRC бот XDCC. Часто помилково називають трекером, через наявність останнього в складі деяких каталогів.

ТоррентРієр

Один із самих популярних движків (російськомовних і не тільки) BitTorrent-трекеру. 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.

Опис.torrent-файлу

Файл метаданих є словником в bencode форматі з розширенням.torrent – використовується в p2p 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.

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

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.

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

- Блок меню.
- Панель інструментів.
- Вікно інформації про завантажуючі та роздавані файли.
- Вікно докладної інформації.
- Список категорій.
- Вікно рядку стану.

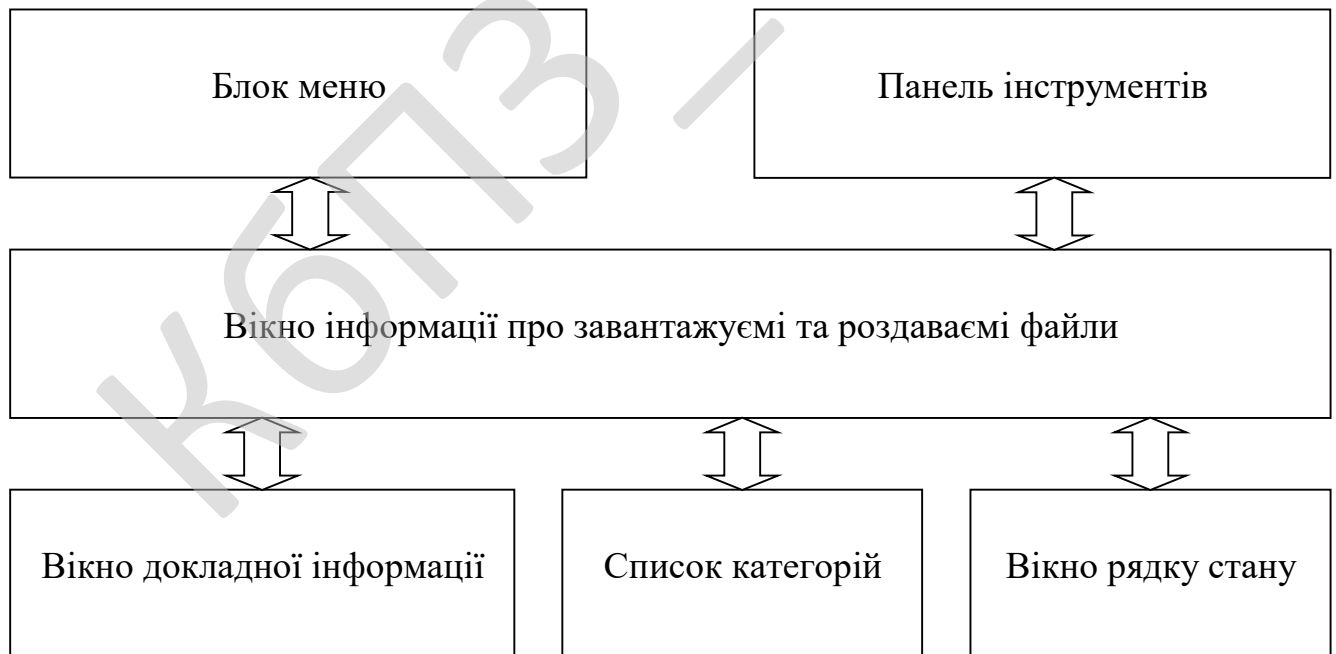


Рисунок 3.1 – Структурна схема системи

3.3 Розробка функціональної схеми

На рисунку 3.2 зображена функціональна схема системи. Нижче розглянемо її більш докладно. Функціональна схема є більш докладним описом структурної схеми. Тому основні блоки співпадають з тими, які були наведені на структурній схемі, але описані більш детально, й приведено їх функціональне призначення.

Основними функціональними блоками є наступні:

1. Блок меню.
2. Панель інструментів.
3. Вікно інформації про завантажувачі та роздавачі файли.
4. Вікно докладної інформації.
5. Список категорій.
6. Вікно рядку стану.

Приведемо їх функціональне призначення:

1. Блок меню:

– 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.

2. Панель інструментів:

- Додати новий торрент.
- Додати торрент з URL.
- Створити новий торрент.
- Перервати завантаження.
- Поставити на паузу завантаження.

- Запустити завантаження.
 - Вгору у черзі.
 - Вниз у черзі.
 - Віддалений доступ.
3. Вікно інформації про завантажуючі та роздавані файли:
- № файлу для завантаження або роздачі.
 - Ім'я файлу який качується або роздається.
 - Розмір файлу.
 - Стан файлу.
 - Швидкість завантаження файлу.
 - Швидкість роздачі файлу.
 - Час завантаження або роздачі.
 - Рейтинг.
4. Вікно докладної інформації:
- Файли.
 - Докладна інформація.
 - Піри.
 - Рейтинги.
 - Трекери.
 - Швидкість.
 - Звіти.
5. Список категорій:
- Завантажуємі.
 - Роздаємі.
 - Готові.
 - Активні.
 - Неактивні.
6. Вікно рядку стану:
- Кількість вузлів DHT.
 - Приймання (швидкість та об'єм).
 - Віддача (швидкість та об'єм).

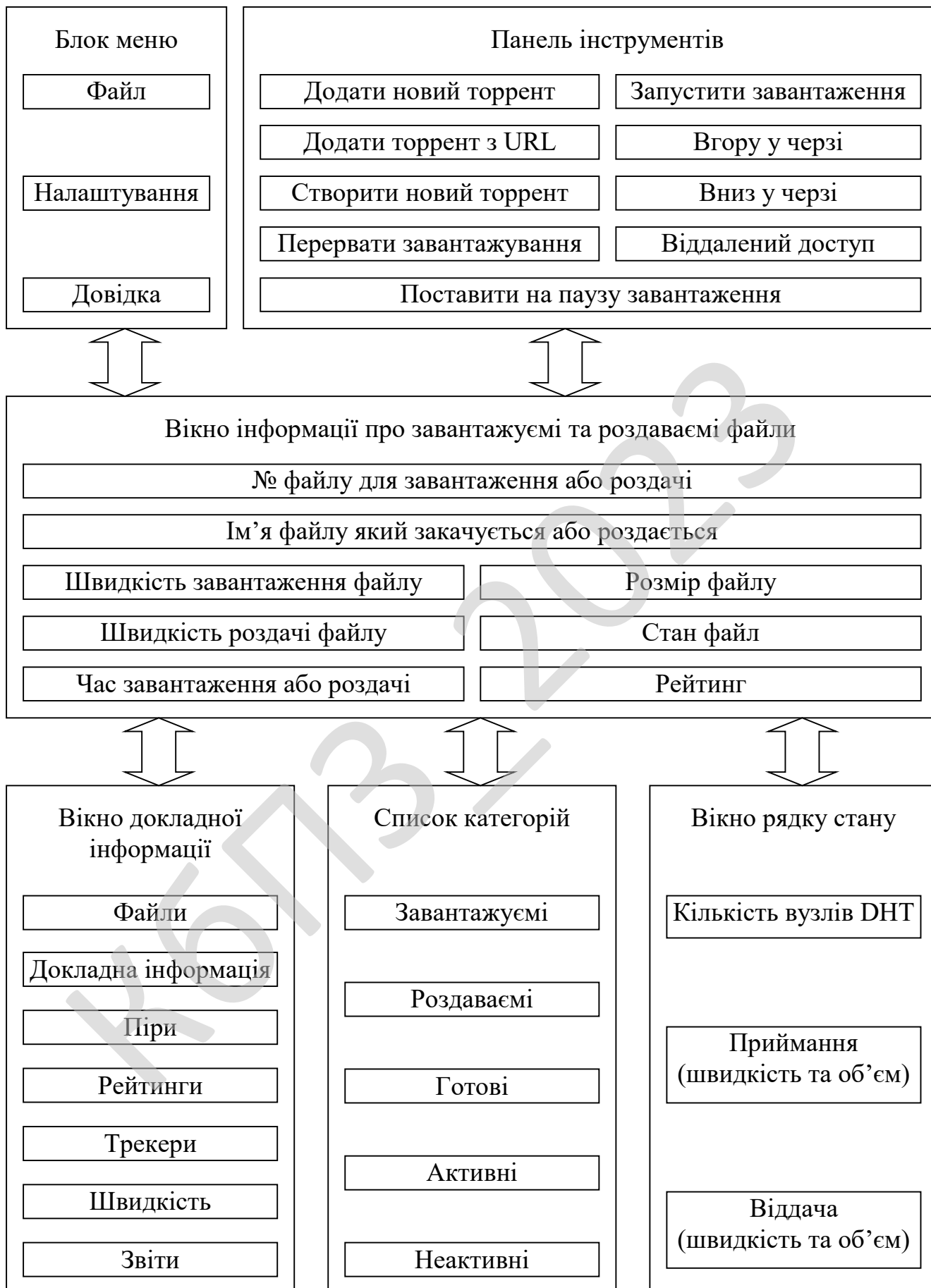


Рисунок 3.2 – Функціональна схема системи

Однією з відмінностей програми, що розроблена в ході виконання магістерського проектування, від інших клієнтів є відсутність авторизації. Також велика увага в ній приділена підтримці різних мов, причому користувач може сам їх редагувати або додавати свої. Для аматорів креативних оформлень пропонується можливість заміни стандартного оформлення на своє власне.

Також однією з особливостей програми, що розроблена в ході виконання магістерського проектування, є мале споживання ресурсів, причому робота самого клієнта ніяким образом не позначається на продуктивності системи.

Інтуїтивно зрозумілий інтерфейс і чіткість структур додатка дозволяють без праці розібратися, як управляти цією програмою.

Після установки додатка в налаштуваннях, викликаних через меню «Налаштування – Конфігурація» (Ctrl+P), можна вибрати самі оптимальні параметри для Вашого комп'ютера й з'єднання Інтернет. Ви можете вказати в яку папку будуть закачуватися файли, обмежити швидкість завантаження й віддачі файлів, вибрати мову, а також створити планувальник завдань.

При виборі вкладки «З'єднання» варто звернути увагу на наявність поставлених прапорів:

- Включити UPnP – переадресацію.
- Включити переадресацію NAT.
- У виключення брандмауера.

Останні два дозволять Вам уникнути такої проблеми, як заборона на вхідні з'єднання, що може негативно відбитися на продуктивності системи. Порт вхідних з'єднань рекомендується вибрати випадковим за допомогою відповідної кнопки. Для тих, у кого стоїть проксі-сервер, є можливість указати необхідні параметри з'єднання.

На вкладці «Швидкість» Ви можете вказати обмеження швидкості для завантаження й віддачі, указавши конкретні значення у відповідних полях. Обмеження швидкості може бути корисним, коли крім торрентів Ви активно використовуєте свій Інтернет або на одному виділеному каналі перебуває


					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

декілька користувачів, що використовують Інтернет. Крім того на цій вкладці можна вибрати значення кількості з'єднань, кількість пірів, приєднаних до одному торренту, і кількість слотів віддачі одного торренту. Кількість з'єднань краще зробити не більшим для оптимізації завантаження, а от кількість пірів навпаки збільшити. Аналогічне збільшення рекомендується зробити й для слотів.

Вкладка BitTorrent дозволяє управляти з'єднанням мережі, а також підтримує DHT-мережу. Вибір однойменних прапорів дозволить трохи збільшити швидкість завантаження, а пункт «Пошук локальних пірів» може трохи розвантажити інтернет-канал за рахунок завантаження інформації з локальних ресурсів. При необхідності можна також включити функцію шифрування протоколу.

Інтуїтивно зрозумілий інтерфейс дозволить розібратися із програмою, що розроблена в ході виконання магістерського проектування, гус без особливих проблем. Докладна інформація й статистика про поточні завантаження виводиться в головному вікні додатка. Кожне закачування супроводжується назвою, розміром файлу, відсотком завантаження, а також інформацією про кількість інших користувачів у мережі, що віддають або закачують подібний файл. Кожне закачування може мати свій статус і відповідно ставитися до того або іншого статусу закачувань:

- Що завантажуються.
- Готові.
- Активні.
- Неактивні.
- Мітка.


Щоб додати нове закачування в клієнт потрібно натиснути кнопку  або в меню вибрати «Файл – Додати торрент...» (Ctrl+O). У вікні, що відкрилося, потрібно вибрати папку, у якій у нас збережені файли-torrent, які попередньо були закачені із трекера. У наступному вікні з інформацією про файл можна вибрати різні налаштування для закачування, такі як: «Додати мітку»,


«Поставити в початок черги», «Не перевіряти геш». Однак всі ці параметри є не обов'язковими й закачування буде відбуватися у звичайному нормальному режимі й без них.

Потім Ваше закачування відразу з'явиться в списку всіх закачувань у головному вікні додатка й клієнт почне шукати в мережі інших комп'ютерів, що мають той же файл, що й у Вас.

Будь-яке закачування можна підняти/опустити в списку пріоритетів закачування, призупинити на час, або зовсім видалити.

Якщо користувачів з таким файлом у мережі близько 1000 і Ваша швидкість інтернету становить не менш 5 Mb/c – будьте впевнені, що файл розміром 700 Mb (стандартний розмір фільму) захитається не більш ніж за 10 хв.

Щоб відкрити папку в яку був заколисаний файл, потрібно два рази клацнути за назвою файлу у вікні закачувань. Але крім закачування файлів з інших комп'ютерів Ви також можете завантажувати в торрент і свої власні й передавати їх безлічі користувачів по усьому світі. Для цього натисніть кнопку  або з меню «Файл – створити новий торрент» (Ctrl + N). Виберіть файл, що хочете опублікувати, а в поле «Трекери» обов'язково введіть хоча б 1 адреса торрент-трекера, наприклад, [http:// free-torrents.org/](http://free-torrents.org/) Інші параметри є опціональними, але рекомендується відразу поставити прапор «Почати роздачу». Варто також помітити, що для завантаження в торрент можна викладати не тільки файли, але й цілі папки, у яких може втримуватися яка завгодно кількість і обсяг файлів. Кнопка «Створити й зберегти» створить на Вашому комп'ютері новий файл торренту.

Ваш новий файл додасться автоматично в список закачувань і буде відзначений червоним символом , що означає, що цей файл закачується безпосередньо з Вашого комп'ютера. Залишилося лише опублікувати файл торренту, що важить не більше 20 Kb, на якому-небудь форумі або відправити його своїм друзям.

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

Для тих, хто за якимись причинами хоче завантажити файл, а потім виключити комп'ютер, але не може цього зробити – існує можливість автоматичного завершення роботи. Для цього в меню «Налаштування – Завершення роботи» необхідно вибрати відповідний пункт.

RSS – функція програми, що розроблена в ході виконання магістерського проектування, що дозволяє автоматично переглядати й завантажувати потоки з ваших улюблених торрент-трекерів. Для використання RSS з меню «Файл – Додати RSS стрічку» у вікні, що відкрилося, потрібно додати URL-адресу стрічки й увести її псевдонім. Автоматична підписка дозволить Вам завжди бути в курсі самих останніх новин або подій без відвідування самого трекеру.

Для відключення живлення RSS з контекстного меню, викликаного правою кнопкою миші за назвою каналу, виберіть «Відключити канал». Інформація більше не буде надходити по цій стрічці. Для включення каналу з того ж контекстного меню виберіть «Включити канал».

Ще однією цікавою функцією програми, що розроблена в ході виконання магістерського проектування, є web-інтерфейс. Для його включення потрібно в налаштуваннях програми вибрати однойменну вкладку й установити прапор «Включити WEB UI». Потім вказати логін, пароль і значення номера порту й IP-адреси, з якого можна буде заходити в цю систему. Після цього скопіюйте в папку %AppData%програми, що розроблена в ході виконання магістерського проектування, rus (просто вставте це посилання в поле адреси на своєму комп'ютері) наступний файл. Потім в адресному рядку браузера введіть наступну адресу: <http://Ваш IP-адреса: номер порту/gui/>. Якщо на екрані нічого не з'явилося – запустите знову клієнт, а в браузері натисніть Ctrl+F5. Якщо всі зробили вірно – побачите той же самий інтерфейс клієнта з усіма закахуваннями й статистикою, що й на звичайному клієнті.

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

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

3.4 Розробка діаграми процесів

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

З рисунку видно, що першим процесом, який завантажується у системі, є процес виведення головного вікна програми. Він взаємодіє з наступними процесами:

- Процес завантаження.
- Процес роздачі.
- Процес встановлення параметрів.

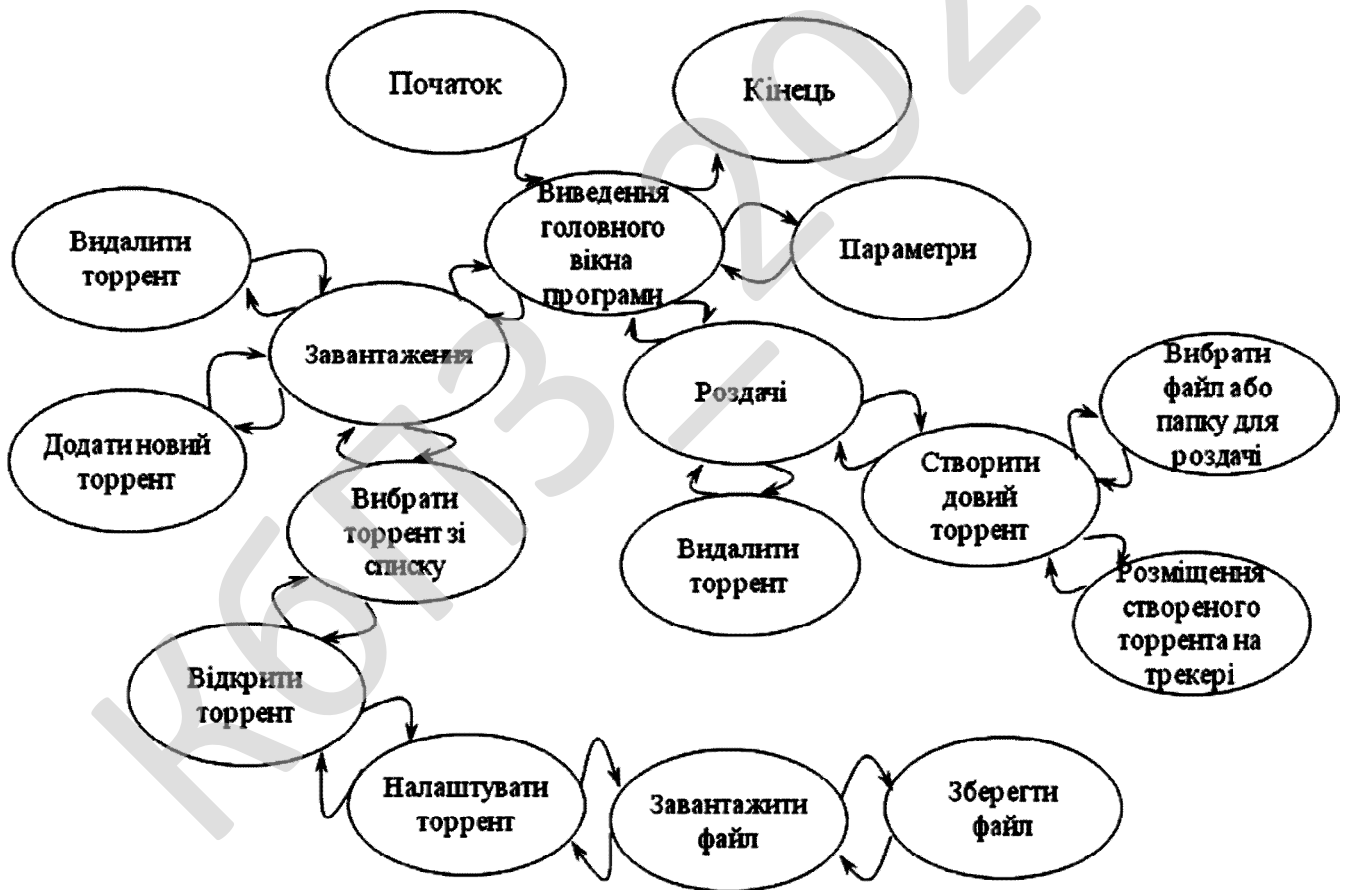


Рисунок 3.3 – Діаграма взаємодії процесів

Процес завантаження взаємодіє з наступними процесами:

- Процес видалення торренту.
- Процес додавання нового торренту.
- Процес вибору торренту зі списку.

Процес вибору торренту зі списку, у свою чергу, взаємодіє з процесом відкриття торренту, який взаємодіє з процесом налаштування торренту.

Процес налаштування торренту взаємодіє з процесом завантаження торренту, який, у свою чергу, взаємодіє з процесом збереження файлу, викачаного з мережі.

Процес роздачі взаємодіє з наступними процесами:

- Процес видалення торренту.
- Процес створення нового торренту.

У свою чергу, процес створення нового торренту взаємодіє з наступними процесами:

- Процес обирання файлу або папки для роздачі.
- Процес розміщення створеного торрента на трекері.

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

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

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. Після цього користувач обирає, яку дію йому робити:

- Завантажити файл з мережі.
- Поставити файл на роздачу.

Якщо користувач обирає завантаження файлу з мережі, тоді програма виконує наступні ітерації:

- Додає потрібний торрент у список завантажень.
- Відкриває торрент.
- Налаштовує торрент.
- Запускає підпрограму завантаження файлу з мережі.
- Зберігає файл.

– Виводить повідомлення про те, що файл успішно завантажено та збережено.

Якщо користувач обирає поставити файл на роздачу, тоді програма виконує наступні ітерації:

- Виводиться вікно створення торрента.
- Обирається файл або папка для роздачі.
- Створюється торрент.
- Розміщується створений торрент на трекері.
- Додається торрент у список роздач.

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

Після виконання усіх вищеперерахованих дій, користувач обирає, працювати йому далі з програмним продуктом, або ні.

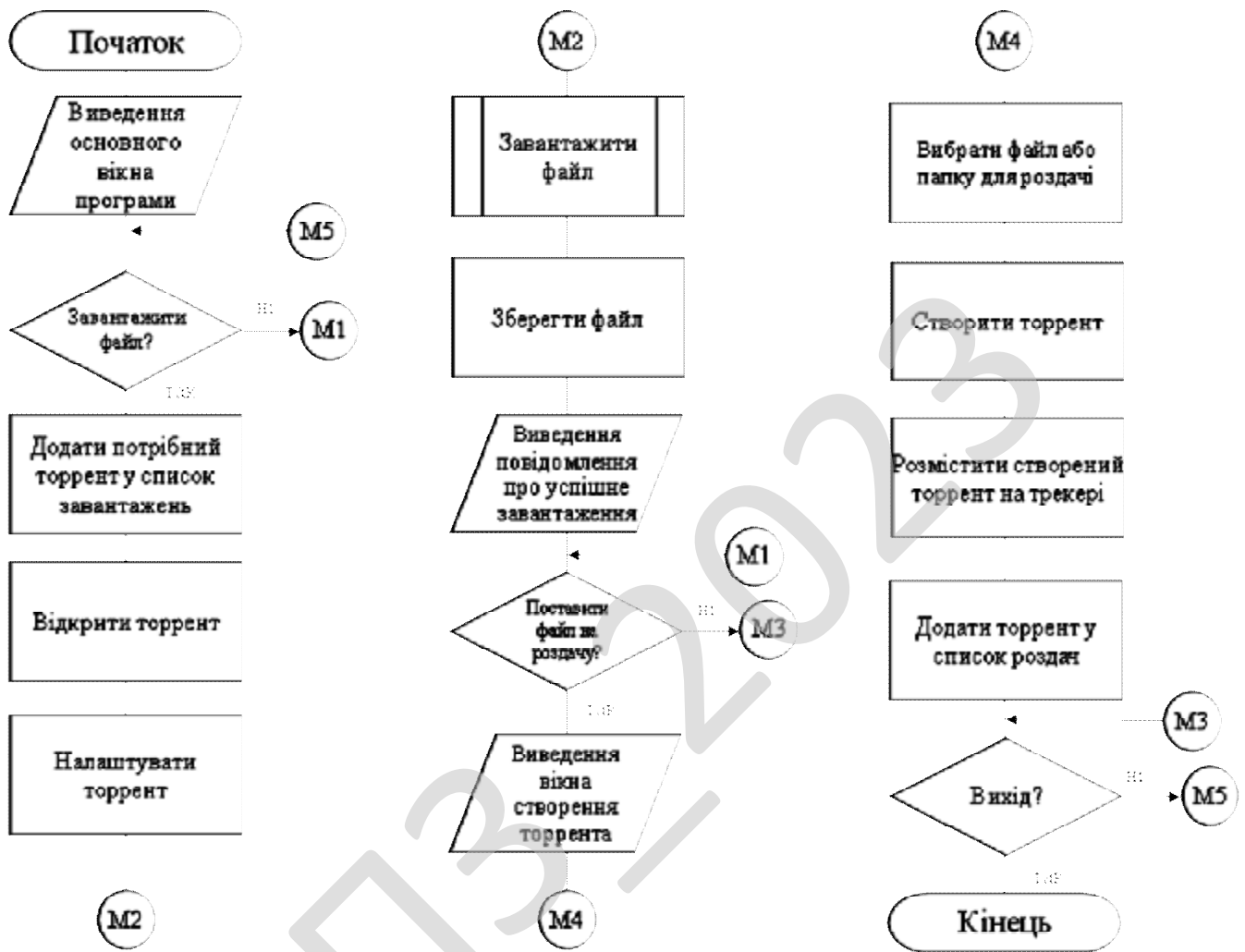


Рисунок 4.1 – Блок-схема основної програми

На рисунку 4.2 зображена блок-схема алгоритму роботи підпрограми завантаження файлу. Вона працює наступним чином.

Спершу відбувається під'єднання до торренту, за адресою вказаною у торрент-файл.

Після цього відправляється трекеру своя адреса.

Наступним кроком є відправлення трекеру геш-суми торрент-файлу.

За цією дією отримується адреси клієнтів, які завантажують, або роздають

цей файл.

Якщо таких адрес немає, тоді програма переходить у режим очікування, доки на трекері не з'являться користувачі, які роздають цей файл.

Якщо ж такі адреси є, тоді виконуються наступні дії:

– Відбувається обмін сегментами між клієнтами торренту.

– Порівнюється контрольна сума отриманого сегменту, з записаною у торрент файлі.

Якщо контрольні суми не співпадають, тоді відбувається повторне завантаження сегменту.

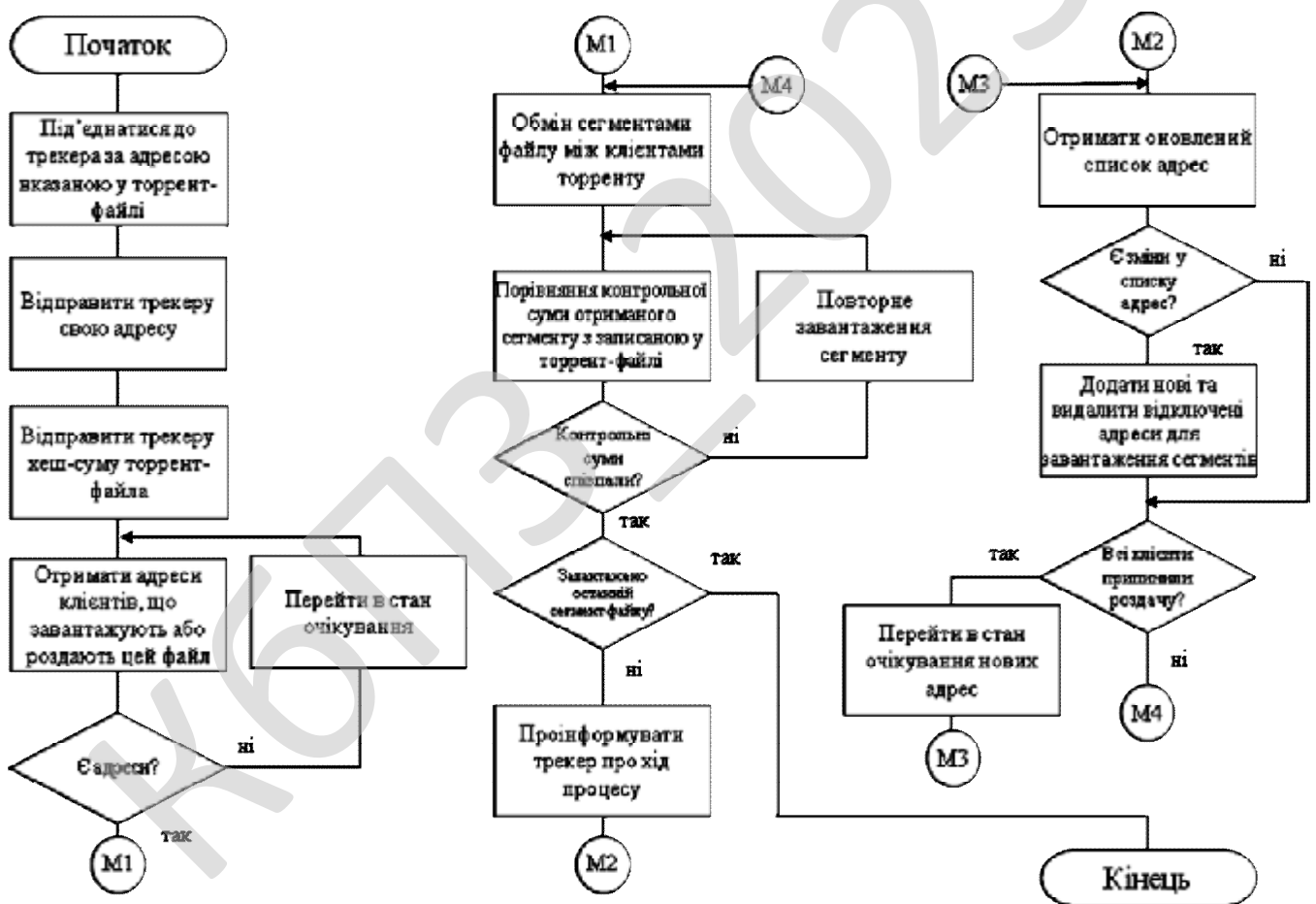


Рисунок 4.2 – Блок-схема алгоритму роботи підпрограми завантаження файлу

Якщо ж контрольні суми співпадають, тоді відбувається перевірка, чи завантажено останній сегмент файлу.

Якщо так, тоді підпрограма завершує свою роботу.

Якщо ні, тоді відбуваються наступні дії:

- Трекер інформується про хід процесу.
- Отримується оновлений список адрес.

Якщо є зміни у списку адрес, тоді додаються нові, та видаляються відключені адреси для завантаження сегментів.

Якщо усі клієнти припинили роздачу, тоді відбувається перехід у стан очікування нових адрес.

На цьому підпрограма закінчує свою роботу.

Наведемо частину коду, яка реалізує вищеперераховані дії.

```
#####
# Class: FileDropTarget
# Щоб вирішити перенесення з подальшим вивільненням для CH_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 : CH_torrent_clientList
# CH_torrent_client список класів, які містять список торрентів
#####
class CH_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]

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 клавіши 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)

```

					БКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

```

        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
    torrentselected = []
    for index in self.getSelected(firstitemonly, reverse):
CH_torrent_clientTorrentTemp = queue.getCH_torrent_clientTorrent(index = index)
        if CH_torrent_clientTorrentTemp is not None:
            torrentselected.append(CH_torrent_clientTorrentTemp)
    return torrentselected
def OnItemSelected(self, event = None):
    selected = self.getTorrentSelected()
    if not selected:
        return
    popupmenu = CH_torrent_clientMenu(self.utility, 'menu_listrightclick')
    # Випадаюче меню.
    if event is None:
        # використовується позиція першого обраного пункту меню(подія при
натисканні клавіши)
        CH_torrent_clientTorrentTemp = selected[0]
        position = self.GetItemPosition(CH_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 : CH_torrent_clientPanel
# Головний CH_torrent_client-клас панелей інтерфейсу користувача
#####
class CH_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 = CH_torrent_clientList(self)
        self.utility.list = self.list
        colSizer.Add(self.list, 1, wx.EXPAND|wx.ALL, 2)
        self.utility.bottomline2 = CH_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 CH_torrent_client на льоту
#####
    def updateColumns(self, force = False):
        # Модифікація показу у колонках для неактивних торрентів
        for CH_torrent_clientTorrentTemp in self.utility.torrents["all"]:

```

					БКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

```

        CH_torrent_clientTorrentTemp.updateColumns(force = force)
#####
# Class : CH_torrent_clientTaskBarIcon
# Іконка таскбара
#####
class CH_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, "CH_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('showCH_torrent_clientwindow'))
        self.utility.actions[ACTION_EXIT].addToMenu(menu, bindto = self)
        return menu
#####
# Class : CH_torrent_clientFrame
# Головний CH_torrent_client Фрейм класу, який містить меню й рядок управління
# та під'єднує CH_torrent_clientPanel
#####
class CH_torrent_clientFrame(wx.Frame):
    def __init__(self, ID, params, utility):
        self.utility = utility
        self.utility.frame = self

```

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

```

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_torrent_client_sb = CH_torrent_clientStatusBar(self)
self.SetStatusBar(self.CH_torrent_client_sb)
try:
    self.SetIcon(self.utility.icon)
except:
    pass
# не модифікуєм GUI
self.GUImодифікація = True
# Запускаємо планувальник перед створенням ListCtrl
self.utility.queue = CH_torrent_clientScheduler(self.utility)
self.window = CH_torrent_clientPanel(self)
# Меню опцій
#####
menuBar = CH_torrent_clientMenuBar(self)
self.SetMenuBar(menuBar)
self.tb = CH_torrent_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_torrent_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_torrent_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 torrents
        for torrent in self.utility.torrents["all"]:
            torrent.updateColumns()
            torrent.updateColor()
#####
# мінімізуємо до трею панель 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

```



```

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_torrent_clientApp
# Головний CH_torrent_client клас додатку під'єднуючий CH_torrent_clientFrame Object
#####
class CH_torrent_clientApp(wx.App):
    def __init__(self, x, params, single_instance_checker, CH_torrent_clientpath):
        self.params = params
        self.single_instance_checker = single_instance_checker
        self.utility = Utility(CH_torrent_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_torrent_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_torrent_client" -
wx.GetUserId())
    if single_instance_checker.IsAnotherRunning():
        #Відправляємо інформацію про торрент до CH_torrent_client
        ClientPassParam(params[0])
    else:
        CH_torrent_clientpath = os.path.abspath(os.path.dirname(sys.argv[0]))
        os.chdir(CH_torrent_clientpath)
        # Почнемо спочатку CH_torrent_client єдиного зразка
        app = CH_torrent_clientApp(0, params, single_instance_checker,
CH_torrent_clientpath)
        app.MainLoop()
if __name__ == '__main__':
    run()
```

4.2 Захист розробленого програмного забезпечення

Дані в програмі захищаються за допомогою використання алгоритму CAST-128 (або CAST5) у криптографії, це блоковий алгоритм симетричного шифрування на основі мережі Фейстеля, який використовується в цілому ряді продуктів криптографічного захисту, зокрема деяких версіях PGP і GPG і крім того схвалений для використання Канадським урядом.

Основні відомості

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

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

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

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

Опис

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

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

CAST-128 належить компанії Entrust Technologies, але є безкоштовним як для комерційного, так і для некомерційного використання. CAST-256 – безкоштовне розширення CAST-128, яке ухвалює розмір ключа до 256 біт і має розмір блоку 128 біт. CAST-256 був одним з первісних кандидатів на AES.

Опис алгоритму

CAST-128 заснований на мережі Фейстеля. Повний алгоритм шифрування викладений у наступних чотирьох кроках:

ВХІД: текст $m_1 \dots m_{64}$, ключ $K = k_1 \dots k_{128}$.

ВИХІД: зашифрований текст $c_1 \dots c_{64}$.

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

Поля заміни

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

Ключі розгорнення

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

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

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

$$z_0z_1z_2z_3 = x_0x_1x_2x_3 \wedge S_5[x_D] \wedge S_6[x_F] \wedge S_7[x_C] \wedge S_8[x_E] \wedge S_7[x_8]$$

$$z_4z_5z_6z_7 = x_8x_9x_{10}x_{11} \wedge S_5[z_0] \wedge S_6[z_2] \wedge S_7[z_1] \wedge S_8[z_3] \wedge S_8[x_A]$$

$$z_8z_9z_{10}z_{11} = x_{12}x_{13}x_{14}x_{15} \wedge S_5[z_7] \wedge S_6[z_6] \wedge S_7[z_5] \wedge S_8[z_4] \wedge S_5[x_9]$$

$$z_{12}z_{13}z_{14}z_{15} = x_{16}x_{17}x_{18}x_{19} \wedge S_5[z_A] \wedge S_6[z_9] \wedge S_7[z_B] \wedge S_8[z_8] \wedge S_6[x_B]$$

$$K_1 = S_5[z_8] \wedge S_6[z_9] \wedge S_7[z_7] \wedge S_8[z_6] \wedge S_5[z_2]$$

$$K_2 = S_5[z_A] \wedge S_6[z_B] \wedge S_7[z_5] \wedge S_8[z_4] \wedge S_6[z_6]$$

$$K_3 = S_5[z_C] \wedge S_6[z_D] \wedge S_7[z_3] \wedge S_8[z_2] \wedge S_7[z_9]$$

$$K_4 = S_5[z_E] \wedge S_6[z_F] \wedge S_7[z_1] \wedge S_8[z_0] \wedge S_8[z_C]$$

$$x_0x_1x_2x_3 = z_8z_9z_{10}z_{11} \wedge S_5[z_5] \wedge S_6[z_7] \wedge S_7[z_4] \wedge S_8[z_6] \wedge S_7[z_0]$$

$$x_4x_5x_6x_7 = z_0z_1z_2z_3 \wedge S_5[x_0] \wedge S_6[x_2] \wedge S_7[x_1] \wedge S_8[x_3] \wedge S_8[z_2]$$

$$x_8x_9x_{10}x_{11} = z_4z_5z_6z_7 \wedge S_5[x_7] \wedge S_6[x_6] \wedge S_7[x_5] \wedge S_8[x_4] \wedge S_5[z_1]$$

$$x_{12}x_{13}x_{14}x_{15} = z_{12}z_{13}z_{14}z_{15} \wedge S_5[x_A] \wedge S_6[x_9] \wedge S_7[x_B] \wedge S_8[x_8] \wedge S_6[z_3]$$

$$K_5 = S_5[x_3] \wedge S_6[x_2] \wedge S_7[x_C] \wedge S_8[x_D] \wedge S_5[x_8]$$

$$K_6 = S_5[x_1] \wedge S_6[x_0] \wedge S_7[x_E] \wedge S_8[x_F] \wedge S_6[x_D]$$

$$K_7 = S_5[x_7] \wedge S_6[x_6] \wedge S_7[x_8] \wedge S_8[x_9] \wedge S_7[x_3]$$

$$K8 = S5[x5] \wedge S6[x4] \wedge S7[xA] \wedge S8[xB] \wedge S8[x7]$$

$$z0z1z2z3 = x0x1x2x3 \wedge S5[xD] \wedge S6[xF] \wedge S7[xC] \wedge S8[xE] \wedge S7[x8]$$

$$z4z5z6z7 = x8x9xAxB \wedge S5[z0] \wedge S6[z2] \wedge S7[z1] \wedge S8[z3] \wedge S8[xA]$$

$$z8z9zAzB = xCxDxExF \wedge S5[z7] \wedge S6[z6] \wedge S7[z5] \wedge S8[z4] \wedge S5[x9]$$

$$zCzDzEzF = x4x5x6x7 \wedge S5[zA] \wedge S6[z9] \wedge S7[zB] \wedge S8[z8] \wedge S6[xB]$$

$$K9 = S5[z3] \wedge S6[z2] \wedge S7[zC] \wedge S8[zD] \wedge S5[z9]$$

$$K10 = S5[z1] \wedge S6[z0] \wedge S7[zE] \wedge S8[zF] \wedge S6[zC]$$

$$K11 = S5[z7] \wedge S6[z6] \wedge S7[z8] \wedge S8[z9] \wedge S7[z2]$$

$$K12 = S5[z5] \wedge S6[z4] \wedge S7[zA] \wedge S8[zB] \wedge S8[z6]$$

$$x0x1x2x3 = z8z9zAzB \wedge S5[z5] \wedge S6[z7] \wedge S7[z4] \wedge S8[z6] \wedge S7[z0]$$

$$x4x5x6x7 = z0z1z2z3 \wedge S5[x0] \wedge S6[x2] \wedge S7[x1] \wedge S8[x3] \wedge S8[z2]$$

$$x8x9xAxB = z4z5z6z7 \wedge S5[x7] \wedge S6[x6] \wedge S7[x5] \wedge S8[x4] \wedge S5[z1]$$

$$xCxDxExF = zCzDzEzF \wedge S5[xA] \wedge S6[x9] \wedge S7[xB] \wedge S8[x8] \wedge S6[z3]$$

$$K13 = S5[x8] \wedge S6[x9] \wedge S7[x7] \wedge S8[x6] \wedge S5[x3]$$

$$K14 = S5[xA] \wedge S6[xB] \wedge S7[x5] \wedge S8[x4] \wedge S6[x7]$$

$$K15 = S5[xC] \wedge S6[xD] \wedge S7[x3] \wedge S8[x2] \wedge S7[x8]$$

$$K16 = S5[xE] \wedge S6[xF] \wedge S7[x1] \wedge S8[x0] \wedge S8[xD]$$

половина, що залишається, ідентична тому, що дане вище, продовження від останнього створило x0..xf, щоб генерувати ключі K17 – K32.

$$z0z1z2z3 = x0x1x2x3 \wedge S5[xD] \wedge S6[xF] \wedge S7[xC] \wedge S8[xE] \wedge S7[x8]$$

$$z4z5z6z7 = x8x9xAxB \wedge S5[z0] \wedge S6[z2] \wedge S7[z1] \wedge S8[z3] \wedge S8[xA]$$

$$z8z9zAzB = xCxDxExF \wedge S5[z7] \wedge S6[z6] \wedge S7[z5] \wedge S8[z4] \wedge S5[x9]$$

$$zCzDzEzF = x4x5x6x7 \wedge S5[zA] \wedge S6[z9] \wedge S7[zB] \wedge S8[z8] \wedge S6[xB]$$

$$K17 = S5[z8] \wedge S6[z9] \wedge S7[z7] \wedge S8[z6] \wedge S5[z2]$$

$$K18 = S5[zA] \wedge S6[zB] \wedge S7[z5] \wedge S8[z4] \wedge S6[z6]$$

$$K19 = S5[zC] \wedge S6[zD] \wedge S7[z3] \wedge S8[z2] \wedge S7[z9]$$

$$K20 = S5[zE] \wedge S6[zF] \wedge S7[z1] \wedge S8[z0] \wedge S8[zC]$$

$$x0x1x2x3 = z8z9zAzB \wedge S5[z5] \wedge S6[z7] \wedge S7[z4] \wedge S8[z6] \wedge S7[z0]$$

$$x4x5x6x7 = z0z1z2z3 \wedge S5[x0] \wedge S6[x2] \wedge S7[x1] \wedge S8[x3] \wedge S8[z2]$$

$$x_8x_9xAxB = z_4z_5z_6z_7 \wedge S_5[x_7] \wedge S_6[x_6] \wedge S_7[x_5] \wedge S_8[x_4] \wedge S_5[z_1]$$

$$xCxDxExF = zCzDzEzF \wedge S_5[xA] \wedge S_6[x_9] \wedge S_7[xB] \wedge S_8[x_8] \wedge S_6[z_3]$$

$$K_{21} = S_5[x_3] \wedge S_6[x_2] \wedge S_7[xC] \wedge S_8[xD] \wedge S_5[x_8]$$

$$K_{22} = S_5[x_1] \wedge S_6[x_0] \wedge S_7[xE] \wedge S_8[xF] \wedge S_6[xD]$$

$$K_{23} = S_5[x_7] \wedge S_6[x_6] \wedge S_7[x_8] \wedge S_8[x_9] \wedge S_7[x_3]$$

$$K_{24} = S_5[x_5] \wedge S_6[x_4] \wedge S_7[xA] \wedge S_8[xB] \wedge S_8[x_7]$$

$$z_0z_1z_2z_3 = x_0x_1x_2x_3 \wedge S_5[xD] \wedge S_6[xF] \wedge S_7[xC] \wedge S_8[xE] \wedge S_7[x_8]$$

$$z_4z_5z_6z_7 = x_8x_9xAxB \wedge S_5[z_0] \wedge S_6[z_2] \wedge S_7[z_1] \wedge S_8[z_3] \wedge S_8[xA]$$

$$z_8z_9zAzB = xCxDxExF \wedge S_5[z_7] \wedge S_6[z_6] \wedge S_7[z_5] \wedge S_8[z_4] \wedge S_5[x_9]$$

$$zCzDzEzF = x_4x_5x_6x_7 \wedge S_5[zA] \wedge S_6[z_9] \wedge S_7[zB] \wedge S_8[z_8] \wedge S_6[xB]$$

$$K_{25} = S_5[z_3] \wedge S_6[z_2] \wedge S_7[zC] \wedge S_8[zD] \wedge S_5[z_9]$$

$$K_{26} = S_5[z_1] \wedge S_6[z_0] \wedge S_7[zE] \wedge S_8[zF] \wedge S_6[zC]$$

$$K_{27} = S_5[z_7] \wedge S_6[z_6] \wedge S_7[z_8] \wedge S_8[z_9] \wedge S_7[z_2]$$

$$K_{28} = S_5[z_5] \wedge S_6[z_4] \wedge S_7[zA] \wedge S_8[zB] \wedge S_8[z_6]$$

$$x_0x_1x_2x_3 = z_8z_9zAzB \wedge S_5[z_5] \wedge S_6[z_7] \wedge S_7[z_4] \wedge S_8[z_6] \wedge S_7[z_0]$$

$$x_4x_5x_6x_7 = z_0z_1z_2z_3 \wedge S_5[x_0] \wedge S_6[x_2] \wedge S_7[x_1] \wedge S_8[x_3] \wedge S_8[z_2]$$

$$x_8x_9xAxB = z_4z_5z_6z_7 \wedge S_5[x_7] \wedge S_6[x_6] \wedge S_7[x_5] \wedge S_8[x_4] \wedge S_5[z_1]$$

$$xCxDxExF = zCzDzEzF \wedge S_5[xA] \wedge S_6[x_9] \wedge S_7[xB] \wedge S_8[x_8] \wedge S_6[z_3]$$

$$K_{29} = S_5[x_8] \wedge S_6[x_9] \wedge S_7[x_7] \wedge S_8[x_6] \wedge S_5[x_3]$$

$$K_{30} = S_5[xA] \wedge S_6[xB] \wedge S_7[x_5] \wedge S_8[x_4] \wedge S_6[x_7]$$

$$K_{31} = S_5[xC] \wedge S_6[xD] \wedge S_7[x_3] \wedge S_8[x_2] \wedge S_7[x_8]$$

$$K_{32} = S_5[xE] \wedge S_6[xF] \wedge S_7[x_1] \wedge S_8[x_0] \wedge S_8[xD]$$

Маскування й перестановка підключів

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

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

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

Змінний розмір ключа

CAST-128 Алгоритм шифрування був розроблений, щоб розмір ключа міг варіюватися від 40 до 128 біт, в 8-бітному кроці (тобто припустимі розміри ключа

рівняються 40, 48, 56, 64..., 112, 120, і 128 бітам). Для змінної роботи розміру ключа специфікація наступні:

1) Для розмірів ключа до й включаючи 80 бітів (тобто, 40, 48, 56, 64, 72, і 80 бітів) алгоритм точно такої ж, але використовує 12 раундів замість 16;

2) Для розмірів ключа більше, чим 80 бітів, алгоритм використовує повні 16 раундів;

3) Для розмірів ключа менше, чим 128 бітів ключ доповнений нульовими байтами (у самих правих, або молодших, позиціях) до 128 биток (тому що розклад ключа CAST 128 ухвалює вхідний ключ 128 бітів).

Розшифрування

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

КБГЗ-2023

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено головне вікно програми. З нього ми бачимо, що інтерфейс користувача складається з наступних блоків:

- Блок меню.
- Блок клавіш швидкого доступу.
- Вікно відображення стану торрентів.
- Вікно відображення загрузки та роздачі файлі через торрент-трекер.

Блок меню складається з наступних елементів:

- Файл.
- Дії.
- Інструменти.
- Параметри.
- Довідка.

Блок клавіш швидкого доступу складається з наступних елементів:

- Додати торрент-файл.
- Додати торрент-файл з локальної папки.
- Додати торрент-файл з мережі.
- Запустити файл на закачку.
- Поставити на паузу.
- Перервати закачку.
- Видалити торрент-файл зі списку трекерів.
- Пошук.

Вікно відображення стану торрентів включає у себе наступні елементи:

- Всі торренти.
- Завантажуємі файли.
- Завантажені за збережені файли.
- Файли, які роздаються.
- Файли, які знаходяться на паузі.

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

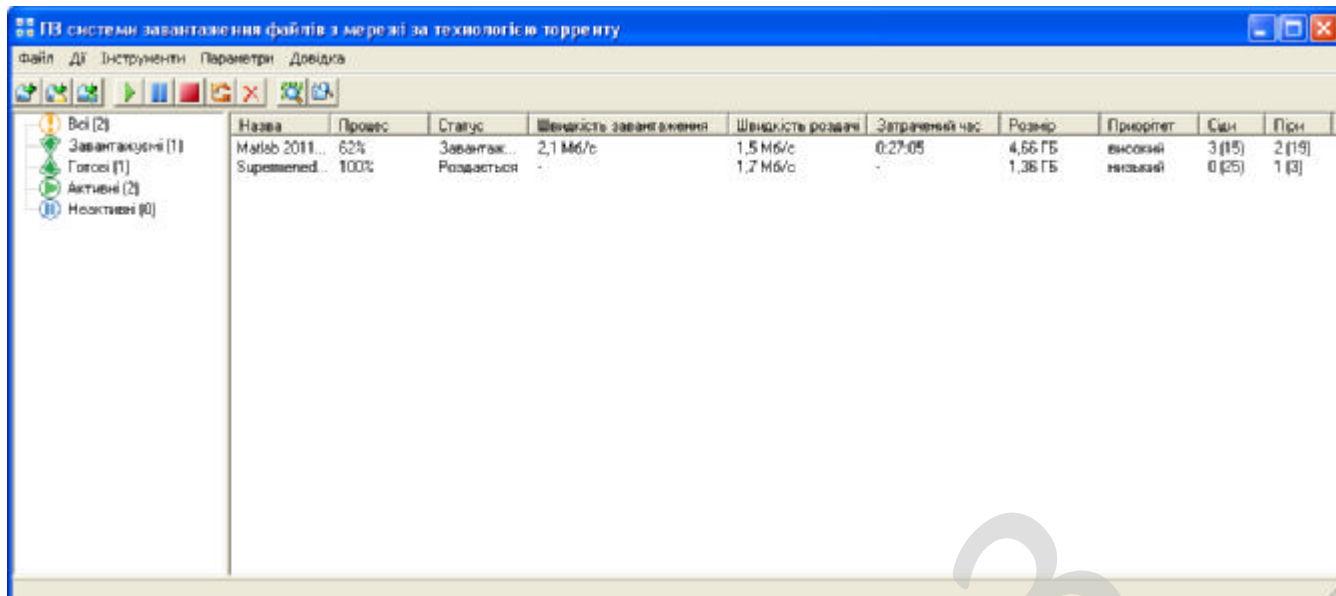


Рисунок 5.1 – Головне вікно програми

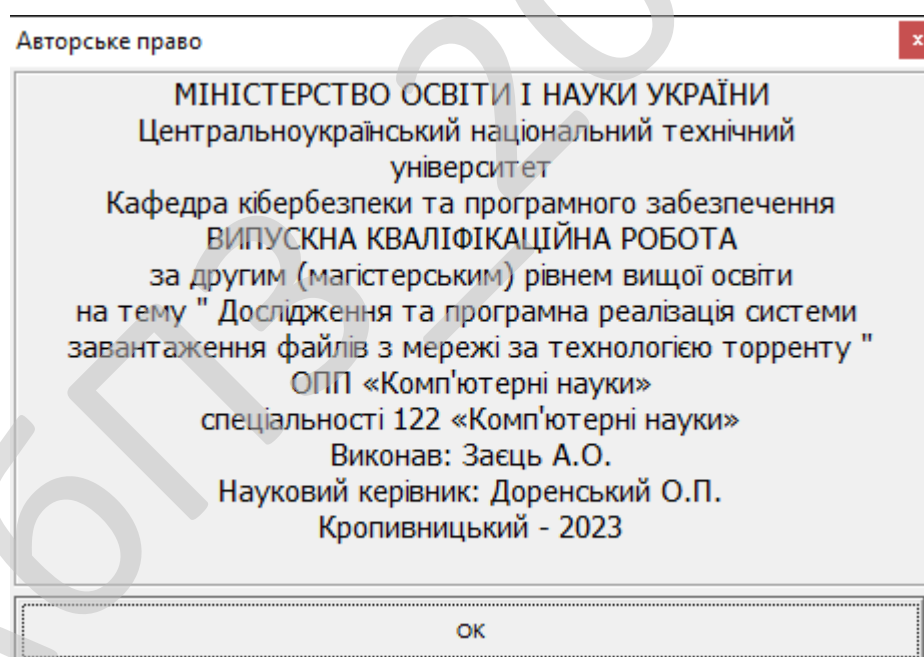


Рисунок 5.2 – Довідка

На рисунку 5.2 зображено вікно довідки, з якого надається інформація, про тему, розробника, керівника магістерського проекту та місце його виконання.

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи завантаження файлів з мережі за технологією торренту.

Метою розробки є дослідження та програмна реалізація системи завантаження файлів з мережі за технологією торренту.

Об'єктом дослідження є процес завантаження файлів з мережі за технологією торренту.

Предметом дослідження є методи завантаження файлів з мережі за технологією торренту.

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

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

- Удосконалено метод завантаження файлів з мережі за технологією торренту.
- Розроблено вітчизняний продукт завантаження файлів з мережі за технологією торренту, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

Продовження таблиці 7.1

1	2	3
7. Кількість макетів вхідної інформації	–	3
8. Кількість форм вихідної інформації.	–	4
9. Мова програмування (1-6)	–	3
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.	–	22000
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-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

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	10	900	15
Монітор	60	10	600	10
Клавіатура	30	10	300	5
Маніпулятор «мишка»	30	10	300	5
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	2	60	1
Кабельні господарства ЛОМ на 1 м.п	2,5	280	700	11,67
Копіювальний апарат	140	1	140	2,33
Усього за рік:			З _ч	53,33

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:

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

$$\Phi_{\text{др}}^c = \frac{53 \cdot 3}{1,2} = 132,5 \text{ год.}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

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

$$Ч_{\text{ел}} = 132,5 / (60 \cdot 8) = 0,27 \text{ ставки.}$$

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

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

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

Посада	Вид роботи	Час	К-ть штатних одиниць
Адміністратор загальної мережі, аналітик	Адміністрування локальної мережі, поштового та серверу DNS (OC FreeBSD), маршрутизатора Cisco, доменного контролеру Windows Server 2012 R2, серверу доступу ADSL (OC Linux), налаштування ADSL, VPN, PPPoE, Frame Relay, Wi-Fi	2	0,5
	Налаштування і конфігурування базової станції безпроводного зв'язку (CMTS)	0,5	
	Розробка та впровадження проектів з організації зв'язку між віддаленими об'єктами, ЛОМ	0,5	
	Забезпечення цілодобової роботи зв'язку клієнтів до мережі Інтернет	1	
Всього		4	

Продовження таблиці 7.4

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

Складемо штатний розклад виконавців.

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньомісячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	1	18132	54396
Продакт-менеджер	0,25	14000	10500
Інженер-програміст	2,8	18000	151200
Інженер-електронщик	0,27	14000	11340
Інженер-системотехнік	0,25	14000	10500
Адміністратор мережі	0,5	12000	18000
Системний програміст	0,25	11500	8625
Дизайнер WEB	0,25	12000	9000
Інженер-верстальник	0,25	11700	8775
Бухгалтер-економіст	0,5	12500	18750
Всього за період розробки	$R_{cn} = 6,32$	-	$\Phi_{роб} = 301086$

Розрахуємо середньоденну зарплату одного виконавця:

$$Z_{сд} = \frac{\Phi_{роб}}{R_{cn} \cdot F_{гр}}, \quad (7.8)$$

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

$$Z_{сд} = \frac{301086}{6,32 \cdot 60} = 794 \text{ грн.}$$

7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника

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

$$B_{уд} = R_{cn}^1 \cdot S_{уд} \cdot U_{пл}, \quad (7.9)$$

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

де: $R_{сн}^1$ – кількість робочих місць виконавців, шт. Приймаємо 8 робочих місць;
 S_y – питома площа на одне робоче місце, m^2 ;
 $Ц_{пл}$ – 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 $8m^2$. 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:

$$I_{нв} = R_{сн}^1 \cdot Ц_{м}, \quad (7.10)$$

де: $Ц_{м}$ – ціна меблів для одного робочого місця, грн.

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

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

Специфікація на обчислювальну техніку наведена в таблиці 7.7.

Дані по оптовій ціні на обладнання та комплектуючі вибирались за пропозицією інтернет ресурсу hotline за 26.10.23 – джерело <https://hotline.ua>

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Персональний комп'ютер		11186
Системний блок		6490
Процесор	Intel Core i7-4770 (4 (8) ядра по 3.4 - 3.6 GHz), 8 MB Smart Cache	-
Системна плата	Intel Q85 Express Chipset, 4x USB 3.0, 6x USB 2.0, 4x Audio, 1x VGA, 2x DisplayPort, 1x COM-порт, 1x LAN (RJ-45), 2x PS/2	-
Відеокарта	nVidia GeForce GT 730, 2 GB GDDR3, 64-bit	-
Жорсткий диск	SSD диск Samsung 870 Evo-Series 250GB 2.5" SATA III V-NAND 3bit MLC (TLC) (MZ-77E250BW)	-
Оперативна пам'ять	DDR3 4GB 1333 MHz CL9 SAMSUNG 1,5V (M378B5273CH0-CH9) два модулі	-
DVD-привод	Super Multi LG SATA DVD±RW R+22x/24x, RW+8x/-6x, DL+16x/-12x, RAM 12xMB SecurDisc, black (GH22NS40RBB)	-
Корпус	HP ProDesk 600 G1 Tower, 320W	-
Кулер	-	-
Кардрідер внутрішній	USB 2.0 Card reader STORM CR-35U1A4-EB int. 3.5", 1*USB2.0+AUDIO+1394, multi: A, Type Cards, black	240
інше	Клавіатура, мишка	-

Таблиця 7.8 – Вартість основних фондів та амортизаційні відрахування розробника

Групи та види основних фондів	Балансова вартість, грн.	Амортизація	
		Норма, %	Відрахування, грн.
1	2	3	4
Група 3			
1. Будівлі	1280000	-	-
2. Передавальні пристрої	128000	-	-
Всього по групі	1408000	5	70400
Група 4			
3. Обчислювальна техніка	214804	-	-
Всього по групі	214804	50	107402
Група 5, 6			
4. Вимірювальні пристрої	5190	25	-
5. Транспортні засоби	0	20	-
6. Господарський інвентар	28000	25	-
Всього по групі	33190	-	8297,5
Нематеріальні активи			
7. Нематеріальні активи	22000	10	2200
Разом	$K_p = 1677994$		$A_p = 188299,5$

Примітка: вартість автомобіля приймаємо рівною нулю.

7.5 Визначення собівартості розробки та ціни програмної продукції

Визначимо основну зарплату виконавців:

$$Z_o = \frac{Z_{\text{ср}} \cdot \Gamma_{\text{мз}}}{N_e}, \quad (7.11)$$

де: N_e – кількість екземплярів програм, шт.

$$Z_o = 794 \cdot 158 / 22 = 5700 \text{ грн.}$$

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, %.

$$Z_d = 5700 \cdot 10 \cdot 0,01 = 570 \text{ грн.}$$

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

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

де: H_c – відрахування на соціальні потреби, %.

$$C_{\text{оц}} = 0,01 \cdot 22(5700 + 570) = 1379 \text{ грн.}$$

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

$$\Gamma_{\text{осп}} = Z_o \cdot H_z \cdot 0,01, \quad (7.14)$$

де: H_z – загальногосподарські витрати, %.

$$\Gamma_{\text{осп}} = 5700 \cdot 15 \cdot 0,01 = 855 \text{ грн.}$$

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

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

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

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

Згідно прийнятих норм на підприємстві n_{sum} приймаємо 0,75 пачки паперу на період розробки. Тоді, враховуючи, що вартість пачки паперу складає $Ц_n=200$ грн., визначаємо вартість паперу за період розробки:

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

$$З_{M1} = 200 \cdot 0,75 = 150 \text{ грн.}$$

Згідно прийнятих норм по комплектації до вартості запам'ятовуючих пристроїв входить вартість CD/DVD дисків. Їх кількість дорівнює кількості коробочних версій запропонованого продукту (приймаємо 10):

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

де: $Ц_{\delta}$ – вартість дисків CD/DVD: CDR box – 27,6 грн./шт., DVD-R box – 32,15 грн./шт.

$$З_{M2} = 10 \cdot 27,6 = 276 \text{ грн.}$$

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

$$З_{M3} = \sum Ц_{\gamma}, \quad (7.18)$$

де: $Ц_{\gamma}$ – 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.

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

$$З_M = (150 + 276 + 1702) / 22 = 97 \text{ грн.}$$

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

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

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

$$O_n = 5700 \cdot 15 \cdot 0,01 = 855 \text{ грн.}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_e = 22$ прим.):

$$A_M = \frac{A_P \cdot N_{шт}}{N_e \cdot 12}, \quad (7.20)$$

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		80

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

$$A_m = 188300 \cdot 3 / (22 \cdot 12) = 2140 \text{ грн.}$$

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

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

$$C_n = 5700 + 570 + 1379 + 855 + 97 + 855 + 2140 = 11596 \text{ грн.}$$

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

Для даного програмного забезпечення рівень рентабельності складає 50%.

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

де: P_n – рівень рентабельності, %.

$$\Pi_p = 0,01 \cdot 50 \cdot 11596 = 5798 \text{ грн.}$$

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	5700
2. Додаткова зарплата виконавців	Z_d	570
3. Відрахування на соціальні потреби	C_{oc}	1379
4. Загальногосподарські витрати	Γ_{ocn}	855
5. Витрати на матеріали	Z_m	97
6. Освоєння нових операційних систем, мов програмування	O_n	855

Продовження таблиці 7.9

1	2	3
7. Амортизація основних фондів	A_m	2140
8. Повна собівартість програмного забезпечення	C_n	11596
9. Плановий прибуток	P_p	5798
10. Ціна підприємства $C_n = C_n + P_p$	C_n	17394
11. Податок на додану вартість $ПДВ = 0.01 \cdot H_{об} \cdot C_n$	$ПДВ$	3478,8
12. Відпускна ціна програмної продукції $C = C_n + ПДВ$	C	20872,8

7.6 Визначення об'єму капітальних вкладень у споживача програмної продукції

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

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

Найменування капітальних вкладень	Сума за варіантами, грн.	
	Базовий	Новий
Вартість програмної продукції	–	20873
Всього капітальних витрат	–	20873

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

$$T_{сп} = \frac{K_H - K_0}{I_0 - I_H}, \quad (7.28)$$

$$T_{сп} = \frac{20873}{80520 - 33400} = 0,44 \text{ року.}$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

7.9 Висновки

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

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		86

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Охорона праці – це: система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини в процесі трудової діяльності;

Охорона праці є складовою частиною безпеки життєдіяльності [3,4].

Законом України “Про охорону праці” [3] регламентуються загальні положення державної політики в галузі охорони праці, а конкретизуються ці положення нормативно-правовими актами про охорону праці, зокрема Наказом Міністерства соціальної політики України 14.02.2018 № 207, який зареєстровано в Міністерстві юстиції України 25 квітня 2018 р. за №508/31960 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» [5], яким затверджено нормативно-правовий акт з охорони праці НПАОП 0.00-7.15-18, «Правила охорони праці під час експлуатації електронно-обчислювальних машин», та «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98.

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

Ці шкідливі фактори можуть привести до професійних захворювань.

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		87

Розглянемо шкідливі чинники роботи програмістів керуючись наступними нормативно-правовими актами: «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98 [5], та «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» НПАОП 0.00-7.15-18.

Умови праці програміста включають наступні фактори:

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

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

Загальний нагляд за додержанням норм охорони праці покладено на прокуратуру, спеціальний покладено на професійні спілки. За безпекою контроль праці здійснюють державні й відомчі спеціалізовані інспекції.

У Законодавстві про працю міститься вимоги і норми з виробничої санітарії, техніки безпеки та норми, що регулюють робочий час, час відпочинку, звільнення та переведення на іншу роботу, а також норми праці щодо жінок, молоді, гігієнічні норми і правила, тощо.

8.2 Аналіз умов праці

Фірма дотримується всіх правил з охорони праці і слідкує за їх дотриманням.

При виконанні робіт на комп'ютерах працівникам необхідно дотримуватись вимог загальної інструкції з охорони праці.

До роботи на комп'ютерах допускаються особи, які пройшли: медичний огляд, навчання по професії, вступний інструктаж з охорони праці та первинний

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		88

інструктаж з охорони праці на робочому місці. В подальшому вони проходять повторні інструктажі з охорони праці на робочому місці один раз на півріччя, періодичні медичні огляди один раз на два роки.

Основним обладнанням робочого місця є монітор, системний блок, миша та клавіатура.

Робочі місця розташовані на відстані не менше 1,5 м від стіни з вікнами, від інших стін на відстані 1 м та між собою на відстані не менше 1,5 м. Відносно вікон робоче місце доцільно розташовувати таким чином, щоб природне світло падало на нього збоку, переважно зліва.

Монітор розташований на робочому місці так, щоб поверхня екрана знаходилася в центрі поля зору на відстані 400-700 мм від очей користувача. Елементи робочого місця розміщуються так, щоб витримувалася однакова відстань очей від екрана, клавіатури, тексту.

Джерела освітлення розташовані з обох боків екрану паралельно напрямку погляду. Для уникнення світлових відблисків екрану, клавіатури в напрямку очей користувача, від світильників загального освітлення або сонячних променів, також використовують антиблікові сітки, спеціальні фільтри для екранів, захисні козирки, на вікнах – жалюзі.

Використовуються скляні поляризаційні фільтри вони забезпечують найкращу якість зображення. Вони усувають практично всі відблиски, роблять зображення чітким і контрастним.

Зручна робоча поза при роботі з комп'ютером забезпечується регулюванням висоти робочого столу, крісла та підставки для ніг. Працівники мають пам'ятки, що раціональною робочою позою вважається положення, при якому ступні працівника розташовані горизонтально на підлозі або підставці для ніг, стегна зорієнтовані у горизонтальній площині, верхні частини рук – вертикальні. Кут ліктьового суглоба коливається в межах 70 – 90°, зап'ястя зігнуті під кутом не більше ніж 20°, нахил голови 15 – 20°.

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		89

– необхідно стійко розташовувати клавіатуру на робочому столі, не опускати її хитання. Під час роботи на клавіатурі сидіти прямо, не напружуватися;

– для забезпечення несприятливого впливу на користувача пристроїв типу ”миша” належить забезпечувати вільну велику поверхню столу для переміщення ”миші” і зручного упору ліктявого суглоба;

– не дозволяються сторонні розмови, подразнюючі шуми;

– періодично при вимкненому комп'ютері прибирати ледь змоченою мильним розчином бавовняною ганчіркою порох з поверхонь апаратури. Екран ВДТ та захисний екран протирають ганчіркою, змоченою у спирті. Не дозволяється використовувати рідинні або аерозольні засоби чищення поверхонь комп'ютера.

Працівники не повинні порушувати правил з охорони праці та їм забороняється:

– самостійно ремонтувати апаратуру. Ремонт апаратури здійснюється спеціалістами з технічного обслуговування комп'ютера, 1 раз на півроку повинні відкривати процесор і вилучати пирососом пил і бруд, що накопичилися;

– класти будь-яку предмети на апаратуру комп'ютера;

– закривати будь-чим вентиляційні отвори апаратури, що може призвести до її перегрівання і виходу з ладу.

Для зняття статичної електрики рекомендується час від часу доторкатися до металевих поверхонь.

Розташувати принтер необхідно поруч з системним блоком таким чином, щоб з'єднувальний шнур не був натягнутий. Забороняється ставити принтери на системний блок.

Для досягнення найбільш чистих, з високою роздільністю зображень і щоб не зіпсувати апарат, має використовуватися папір, вказаний в інструкції до принтера. При змінанні паперу потрібно відкрити кришку і обережно витягнути лоток з папером.

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

Працівника потрібно дотримуватися вимоги безпеки після закінчення роботи:

- закінчити та записати у пам'ять комп'ютера файл, що знаходиться в роботі;
- вимкнути принтер та інші периферійні пристрої. Штепсельні вилки витягнути з розеток. Накрити клавіатуру кришкою запобігання попаданню в неї пилу;
- прибрати робоче місце;
- ретельно вимити руки теплою водою з милом;
- вимкнути кондиціонер, освітлення і загальне електроживлення;
- пройти в спеціально обладнаному приміщенні сеанс психофізіологічного розвантаження і зняття втоми з виконанням спеціальних вправ аутогенного тренування.

8.4 Пожежна безпека

Пожежі в приміщеннях з оргтехнікою становлять особливу небезпеку, бо поєднані з великими матеріальними збитками. Пожежа може виникнути при взаємодії горючих речовин і джерел запалювання. Горючими речовинами є будівельні та опоряджувальні матеріали, пластмасові корпуси техніки, шнури тощо. Джерелами запалювання можуть бути електронні схеми комп'ютерів, принтерів, пристроїв електроживлення, де внаслідок різних порушень виникає перегрівання елементів, утворюються електричні іскри та дуги, здатні спричинити займання горючих матеріалів.

З метою виявлення початкової стадії займання необхідно використовувати пристрої систем автоматичного пожежогасіння там, де цього вимагають правила пожежної безпеки.

При обслуговуванні, ремонтних та профілактичних роботах використовуються різні легкозаймисті рідини, прокладаються тимчасові

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

електроустаткування або порушення заземлення треба негайно повідомити про це свого начальника для вжиття заходів щодо усунення несправності. Проводити самому ремонт електроустаткування забороняється.

8.5 Розрахункова частина

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

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

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

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

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

N – кількість працюючих.

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

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

де V – об'єм приміщення, м^3 .

Визначаємо вільний об'єм приміщення

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

Питомий вільний об'єм складає

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

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

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		95

Висновки до розділу

У даному розділі магістерської роботи проведено аналіз умов працівника робота якого пов'язана з комп'ютерною технікою. Проведено аналіз основних санітарно – гігієнічних показників в заданому приміщенні, де працівник зайнятий постійною роботою за комп'ютером.. Створені умови повинні забезпечувати комфортну роботу. На підставі вивченої літератури з даної проблеми, були зазначені оптимальні параметри мікроклімату, освітлення, допустимі рівні шуму та іонізуючого випромінювання при роботі з ПЕОМ, а також розраховано найменшу необхідну кількість повітря для вентиляції.

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

КБГІЗ-2023

					VKPM-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		96

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи завантаження файлів з мережі за технологією торренту.

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 завантаження файлів з мережі за технологією торренту.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем завантаження файлів з мережі за технологією торренту.

– Досліджена система завантаження файлів з мережі за технологією торренту.

– На основі отриманих результатів досліджень створена програмна реалізація системи завантаження файлів з мережі за технологією торренту.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання завантаження файлів з мережі за технологією торренту.

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

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

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

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 CAST-128.

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

Розроблена програма має реальний економічний ефект від її впровадження у виробництво у сумі 41902 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 0,44 роки.

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		98

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Заєць А.О. Дослідження та програмна реалізація системи завантаження файлів з мережі за технологією торренту // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2023.
2. Сузи Р. Python. ISBN: 5-94157-097-X. 768 С.
3. Wendell Odom. «CCNA 200-301 Official Cert Guide, Volume 2 Premium Edition eBook and Practice Test». Cisco Press. 2020. – 624 p.
4. Scott Jernigan «CompTIA Network+ Certification All-in-One Exam Guide, Eighth Edition». 2022. – 976 p.
5. Doug Lowe «Networking For Dummies 12th Edition». 2020. – 480 p.
6. Ramon Nastase «Computer Networking: The Beginner's guide for Mastering Computer Networking, the Internet and the OSI Model». 2018. – 186 p.
7. Russ White & Ethan Banks «Computer Networking Problems and Solutions: An Innovative Approach to Building Resilient, Modern Networks». 2017. – 832 p.
8. 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.
9. 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.
10. 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.

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		99

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

12. 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.

13. 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.

14. 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.

15. 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.

16. 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.

17. 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.

18. 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.

19. 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.

20. 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.

21. 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.

22. 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.

23. 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.

24. 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.

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

					БКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		101

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

26. 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.*

27. 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.

28. 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.

29. 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.

30. 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.

31. 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.

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		102

32. 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.

33. 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

34. 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.

35. 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.

36. 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.

37. 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.

38. 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.

39. 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.

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		103

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

41. 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

42. 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.

43. 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.

44. 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.

45. 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.

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

47. 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.

48. 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.

49. 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.

50. 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.

51. Смірнов О.А., Євсєєв С.П., Жукарев В.Ю., Король О.Г., Сорокін В.Є., Мелешко Є.В. Технології і стандарти комп'ютерних мереж. Навчальний посібник для студентів вищих навчальних закладів напрямів підготовки 8.050102 «Комп'ютерна інженерія» та 8.0925 «Автоматизація й комп'ютерно-інтегровані технології». За ред. О.А. Смірнова Гриф «Навчальний посібник» надано у відповідності з листом Міністерства освіти і науки, молоді та спорту України від 1.12.2011 року № 1/11-11258. – Кіровоград: КНТУ 2012. – 454 с

					ВКРМ-122.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		105

Додаток А
(обов'язковий)

Технічне завдання

Зміст

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

					ВКРМ-122.23.0008.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Засць А.О.				Дослідження та програмна реалізація системи завантаження файлів з мережі за технологією торренту	Літ.	Аркуш	Аркушів
Перевірів	Доренський О.П.					М	1	6
Н. Контр.	Коваленко А.С.				ЦНТУ КН-22М-1			
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи завантаження файлів з мережі за технологією торренту.

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. № 32-13 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 завантаження файлів з мережі за технологією торренту.

4 Джерела розробки

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

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

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

					ВКРМ-122.23.0008.00.00.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 завантаження файлів з мережі за технологією торренту;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

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

5.4 Вимоги до архітектури

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

5.5 Вимоги до надійності

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

					ВКРМ-122.23.0008.00.00.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-122.23.0008.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

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

7 Економічні вимоги

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

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

8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинен бути розглянутий аналіз умов праці.

					ВКРМ-122.23.0008.00.00.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 – 105 аркушів.

10 Етапи розробки

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

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

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

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Розрахунок з техніко-економічного обґрунтування.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

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

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 11.12.2023 р.

					ВКРМ-122.23.0008.00.00.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

_____ Доренський О.П.

***Дослідження та програмна реалізація
системи завантаження файлів з мережі за технологією торренту***

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 78

Літера: РП

Кропивницький – 2023 року

CH_torrent_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_torrent_client.Toolbars.toolbars import CH_torrent_clientBottomBar2,
CH_torrent_clientStatusBar, CH_torrent_clientMenuBar, CH_torrent_clientToolBar
from CH_torrent_client.GUI.menu import CH_torrent_clientMenu
from CH_torrent_client.Scheduler.scheduler import CH_torrent_clientScheduler

from webservice import WebListener

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

from CH_torrent_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_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 : CH_torrent_clientList
#
# CH_torrent_client список класів, які містять список торрентів
#
#####
class CH_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_DL_SIZE,
              COL_UL_SIZE,
              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 getTorrentSelected(self, firstitemonly = False, reverse = False):
    queue = self.utility.queue

    torrentsselected = []
    for index in self.getSelected(firstitemonly, reverse):
        CH_torrent_clientTorrentTemp =
queue.getCH_torrent_clientTorrent(index = index)
        if CH_torrent_clientTorrentTemp is not None:
            torrentsselected.append(CH_torrent_clientTorrentTemp)
    return torrentsselected

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

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

    # Випадаюче меню.
    if event is None:
        # використовується позиція першого обраного пункту меню(подія при
натисканні клавіши)
        CH_torrent_clientTorrentTemp = selected[0]
        position =
self.GetItemPosition(CH_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 : CH_torrent_clientPanel
#
# Головний CH_torrent_client клас панелей інтерфейсу користувача
#
#####
class CH_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 = CH_torrent_clientList(self)
self.utility.list = self.list

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

self.utility.bottomline2 = CH_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 CH_torrent_client на льоту
#####
def updateColumns(self, force = False):
    # Модифікація показу у колонках для неактивних торрентів
    for CH_torrent_clientTorrentTemp in self.utility.torrents["all"]:
        CH_torrent_clientTorrentTemp.updateColumns(force = force)

#####
#
# Class : CH_torrent_clientTaskBarIcon
#
# Іконка таскбара
#
#####
class CH_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, "CH_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('showCH_torrent_clientwindow'))
        self.utility.actions[ACTION_EXIT].addToMenu(menu, bindto = self)
        return menu

#####
#
# Class : CH_torrent_clientFrame
#
# Головний CH_torrent_client Фрейм класу, який містить меню й рядок управління
# та під'єднує CH_torrent_clientPanel
#
#####
class CH_torrent_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_torrent_client_sb = CH_torrent_clientStatusBar(self)
        self.SetStatusBar(self.CH_torrent_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_torrent_clientScheduler(self.utility)

        self.window = CH_torrent_clientPanel(self)

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

        self.tb = CH_torrent_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_torrent_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_torrent_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.GUImодифікація = update

    if self.GUImодифікація and not oldval:
        # Force an модифікація of all torrents
        for torrent in self.utility.torrents["all"]:
            torrent.updateColumns()
            torrent.updateColor()

#####
# мінімізуємо до трею панель 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_torrent_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_torrent_clientquitting = True
self.GUIмодифікація = False

# Закриваємо створений торрент
self.utility.actions[ACTION_MAKETORRENT].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_torrent_clientApp
#
# Головний CH_torrent_client клас додатку під'єднуючий CH_torrent_clientFrame
Object
#
#####
class CH_torrent_clientApp(wx.App):
    def __init__(self, x, params, single_instance_checker,
CH_torrent_clientpath):
        self.params = params
        self.single_instance_checker = single_instance_checker

self.utility = Utility(CH_torrent_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_torrent_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_torrent_client" + wx.GetUserId())

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

        # Почнемо спочатку CH_torrent_client єдиного зразка
        app = CH_torrent_clientApp(0, params, single_instance_checker,
CH_torrent_clientpath)
        app.MainLoop()

if __name__ == '__main__':
    run()

```

CH_torrent_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_torrent_client_Engine
#
# Двигун торренту.
#
#####
class CH_torrent_client_Engine:
    def __init__(self, torrent, myid):
        self.torrent = torrent
        self.queue = torrent.queue
        self.utility = torrent.utility

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

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.torrent.metainfo
hash = unhexlify(self.torrent.infohash)

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

self.workarounds = { 'hasexternal': False }

self.color = 'color_startup'

btconfig = self.utility.getBTParams()

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

self.dow = BT1Download(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.torrent.status.value == STATUS_HASHCHECK) and
        self.utility.config.Read('fastresume', "boolean")):
        self.dow.appdataobj.deleteTorrentData(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.torrent)

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

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

    # Встановлюємо прапорець
    if self.torrent.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.torrents["active"][self.torrent]
    except:
        pass

    # Відключаємо відображення часу закачування
    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.torrent)

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.torrent.files.dest

def done(self, event = None):
    self.torrent.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.torrent.status.completed = self.seed

    # оновлюємо дату кожні 20 хвилин
    #####
    if self.utility.config.Read('scrape', "boolean"):
        self.torrent.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.torrent.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.torrent.files.updateProgress()
        self.totalspeed = float(statistics.torrentRate)
        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.torrent.status.isDoneUploading():
            self.TerminateUpload()

    # Модифікація текстових рядків
    self.torrent.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.torrent.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.torrent.files.updateFileProgress(statistics)

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

if self.torrent.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.torrent.actions.oldstatus
    if not oldstatus in activevalues:
        self.shutdown()
        return
    else:
        self.torrent.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.torrent.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 поступаючі 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.torrent.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.torrent.files.getSpaceNeeded()

#
if spaceneeded == 0L:
    return

```

```

dest = self.torrent.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.torrent])

#
# Дивимося, скільки необхідно часу для завантаження
#
def CheckTimeouts(self):
    # Перевіряємо, щоб поглянути якщо нам необхідно перевірити час
    if not self.torrent.connection.timeout or self.torrent.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.torrent.dialogs.details
    if detailwin is None or not detailwin.update:
        return
    detailpanel = detailwin.detailPanel

    if statistics is not None:
        detailpanel.updateFromCH_torrent_clientTorrent()

    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.torrent.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.torrent.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.torrent.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_DL_SIZE:
    text = self.utility.size_format(float(spew[line]['dtotal']))

elif colid == SPEW_UL_SIZE:
    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_PEEPSPEED:
    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 errmsg(self, errmsg):
    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 = errmsg.lower().find(error)
    if index != -1:
        oldlen = len(error)
        errmsg = errmsg[:index] + errors[error] + errmsg[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.torrent.changeMessage(errormsg, "error")

# якщо помилка з'єднання з трекером в параметрі 'timeouttracker'
вказуємо кількість хвилин
#####
if self.torrent.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.torrent.prio
    if currentprio < 4: #пріоритет не нижчий
        self.torrent.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.torrent.connection.seedingtime += timelapse

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

```

```

        newseedingtimeleft = required / self.meanrate['up']
        delta = max(newseedingtimeleft/10, 2)
        if abs(self.torrent.connection.seedingtimeleft -
newseedingtimeleft) > delta:
            # If timer value deviates from theoretical value by more
then 10%, reset it to theoretical value
            self.torrent.connection.seedingtimeleft = newseedingtimeleft
        else:
            # Keep on timing
            self.torrent.connection.seedingtimeleft -= timelapse
            if self.torrent.connection.seedingtimeleft < 0.1:
                self.torrent.connection.seedingtimeleft = 0.1
    else:
        # Set to 366 days (= infinite)
        self.torrent.connection.seedingtimeleft = 999999999999999

    self.seedingtimelastcheck = now

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

    self.torrent.connection.stopEngine()

    self.queue.updateAndInvoke()

def finished(self):
    self.seed = True

    # процес сїда
    self.torrent.status.completed = True
    self.progress = 100.0
    self.torrent.files.updateProgress()

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

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

    self.torrent.updateColor()

    self.queue.updateAndInvoke()

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

# Тільки черга, якщо інші закачки чекають

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

    nexttorrent = inactivetorrents[0]

    # See if this torrent would be started if queued
    queuethis = False
    if (nexttorrent.prio < self.torrent.prio):

```

```
        queuethis = True
    elif (nexttorrent.prio == self.torrent.prio) and
(nexttorrent.listindex < self.torrent.listindex):
        queuethis = True

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

K6П3 - 2023


```
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()
```

КБПЗ - 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 == "Automatic") 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_torrent_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 торрента копіюємо до .torrent
        # 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 getTorrents(self, info = "", infohash_list = None):
    torrents = None

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

    torrents = []

    for infohash in infohash_list:
        torrent = self.utility.queue.getCH_torrent_clientTorrent(info_hash =
infohash)

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

    return torrents

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,
                  "dlsizе"       : COL_DLSize,
                  "ulsizе"       : 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_torrent_clientTorrentTemp in self.utility.torrents["all"]:
            retmsg += CH_torrent_clientTorrentTemp.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.addtorrents.AddTorrentURL(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 OK")
        else:
            if not self.utility.webconfig.Read('allow_delete', "boolean") !=
"1":
                conn.send("Повертає \n помилку = DELETE,Відхилений дозвіл")
                return

            torrents = self.getTorrents(info)
            if torrents is None:
                return

            self.utility.actionhandler.procREMOVE(torrents)
            conn.send("Повертає \n OK")

    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:
        torrents = self.getTorrents(info)
        if torrents is None:
            return

        self.utility.actionhandler.procRESUME(torrents)
        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:
        torrents = self.getTorrents(info)
        if torrents is None:
            return

        self.utility.actionhandler.procSTOP(torrents)
        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":
        torrents = self.utility.torrents["active"].keys()
    else:
        torrents = self.getTorrents(info)
        if torrents is None:
            return

    self.utility.actionhandler.procPAUSE(torrents, 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":
        torrents = self.utility.torrents["all"]
    else:
        torrents = self.getTorrents(info)
        if torrents is None:
            return

    self.utility.actionhandler.procQUEUE(torrents)
    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]
torrents = self.getTorrents(infohash_list = hashlist)
if torrents is None:
    return

for torrent in torrents:
    torrent.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_torrent_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_torrent_client!")
                conn.close()
                continue

            if idkey != "ID":
                conn.send("Повертає \n помилку = Вам потрібні унікальні ID
до управління CH_torrent_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_torrent_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

```

```
cmd) conn.send("Повертає \n помилку = Команда не знайдена: " +  
        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_torrent_client з веб сервісом"

        # якщо запит - не запит з'єднання
        # тоді цей запит торренту записуємо до .torrent
        # 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|torrenturlthatworks")
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 "-----"

```

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_bt1 import defaults as BTDefaults
from BitTornado.parseargs import parseargs
from BitTornado.zurllib import urlopen

from CH_torrent_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 «клеювий» клас, який містить зазвичай функції допомоги і допомагає
# тримати трек об'єктів
#
#####
class Utility:
    def __init__(self, CH_torrent_clientpath):
        self.CH_torrent_clientpath = CH_torrent_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.MakeTorrentDir()

        self.setupWebConfig()

        self.setupTorrentMakerConfig()

        self.setupTorrentList()

        self.torrents = { "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_torrent_clientquitting = False
# self.CH_torrent_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',
        'removetorrent': '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': '000150000',
'color_stripe': '245245245',
'display_interval': '0.8',
'listfont': '',
'diskfullthreshold': '0',
# 'showmenuicons': '1',
'icons_toolbarbottom': [ACTION_MOVEUP,
                        ACTION_MOVEDOWN,
                        ACTION_MOVETOP,
                        ACTION_MOVEBOTTOM,

```

```

        -1,
        ACTION_CLEARCOMPLETED,
        -1,
        ACTION_PAUSEALL,
        ACTION_STOPALL,
        ACTION_UNSTOPALL,
        -1],
'icons_toolbartop': [ACTION_ADDTORRENT,
        ACTION_ADDTORRENTNONDEFAULT,
        ACTION_ADDTORRENTURL,
        -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_torrent_client.conf")
    self.config = ConfigReader(configfilepath, "CH_torrent_client",
defaults)
#     self.config = ConfigReader(configfilepath, "CH_torrent_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_torrent_client/Webservice", defaults)

    def setupTorrentMakerConfig(self):
        defaults = {
            'piece_size': '0',
            'comment': '',
            'created_by': '',
            'announcedefault': '',
            'announcehistory': '',
            'announce-list': '',
            'httpseeds': '',
            'makehash_md5': '0',
            'makehash_crc32': '0',
            'makehash_shal': '0',
            'startnow': '0',
            'savetorrent': '2'
        }

        torrentmakerconfigfilepath = os.path.join(self.getConfigPath(),
"maker.conf")
        self.makerconfig = ConfigReader(torrentmakerconfigfilepath,
"CH_torrent_client/TorrentMaker", defaults)

    def setupTorrentList(self):
        torrentfilepath = os.path.join(self.getConfigPath(), "torrent.list")
        self.torrentconfig = ConfigReader(torrentfilepath, "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_torrent_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_torrent_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 MakeTorrentDir(self):
    torrentpath = os.path.join(self.getConfigPath(), "torrent")
    pathexists = os.access(torrentpath, os.F_OK)
    # Якщо директорія торренту не існує, створюємо її
    if not pathexists:
        os.mkdir(torrentpath)

```

```

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

```

CH_torrent_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_torrent_client.GUI.menu import MenuDialog
from Utility.configreader import ConfigReader
from Utility.constants import * #IGNORE:W0611

#####
#
# Class: CH_torrent_client_optionPanel
#
# Основна структура для панелі вікна опцій
#
# Додаємо кнопку для "Відновити типові значення"
# внизу кожної панелі
#
#####
class CH_torrent_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_torrent_client_optionPanel):
    def __init__(self, parent, dialog):
        CH_torrent_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_torrent_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.otsameip = wx.CheckBox(self, -1,
self.utility.lang.get('security'))
        sizer.Add(self.otsameip, 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.otsameip.SetValue(Read('otsameip', "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('otsameip', self.otsameip.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_torrent_client_optionPanel):
    def __init__(self, parent, dialog):
        CH_torrent_client_optionPanel.__init__(self, parent, dialog)
        sizer = self.sizer

        #
        # Число одночасних активних потоків
        #
        activesection_title = wx.StaticBox(self, -1,
self.utility.lang.get('activetorrents'))
        activesection = wx.StaticBoxSizer(activesection_title, wx.VERTICAL)

        self.numsimtext = wx.SpinCtrl(self, size = wx.Size(60, -1))
        self.numsimtext.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.numsimtext, 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('trignexttorrent')), 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_lineinbox = wx.BoxSizer(wx.HORIZONTAL)
        autostart_lineinbox.Add(self.autostart, 0, wx.ALIGN_CENTER_VERTICAL)
        autostart_lineinbox.Add(self.autostartthreshold, 0,
wx.ALIGN_CENTER_VERTICAL|wx.LEFT, 5)
        autostart_lineinbox.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.UpdateRunningTorrentCounters()

#####
#
# Class: DisplayPanel
#
# Містить установки, як виглядає CH_torrent_client
#
#####
class DisplayPanel(CH_torrent_client_optionPanel):
    def __init__(self, parent, dialog):
        CH_torrent_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_torrent_clientTorrentTemp in self.utility.torrents["all"]:
            CH_torrent_clientTorrentTemp.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_torrent_client_optionPanel):
    def __init__(self, parent, dialog):
        CH_torrent_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_torrent_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.rtd.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_torrent_client/language")
            if config.Exists('language'):
                self.utility.languages[config.Read('language')] = filename

#####
#
# Class: DiskPanel
#
# Містить установки, пов'язані із збереженням файлів
#
#####
class DiskPanel(CH_torrent_client_optionPanel):
    def __init__(self, parent, dialog):
        CH_torrent_client_optionPanel.__init__(self, parent, dialog)
        sizer = self.sizer

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

```

```

sizer.Add(self.torrentbackup, 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.torrentbackup.SetValue(Read('removetorrent', "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('removetorrent',
self.torrentbackup.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_torrent_client_optionPanel):
    def __init__(self, parent, dialog):
        CH_torrent_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 трекера
        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_torrent_client_optionPanel):
    def __init__(self, parent, dialog):
        CH_torrent_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('l_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.prioritizelocal = wx.CheckBox(self, -1,
self.utility.lang.get('prioritizelocal'))
sizer.Add(self.prioritizelocal, 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.prioritizelocal.SetValue(Read('prioritizelocal', "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('prioritizelocal',
self.prioritizelocal.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_torrent_client_optionPanel):
    def __init__(self, parent, dialog):
        CH_torrent_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.cbratio, 0, wx.ALIGN_CENTER_VERTICAL|wx.LEFT, 5)
percent_sizer.Add(wx.StaticText(self, -1, "%"), 0,
wx.ALIGN_CENTER_VERTICAL)

5)
continuesection.Add(percent_sizer, 0, wx.ALIGN_CENTER_VERTICAL|wx.ALL,

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.cbratio.SetValue(Read('uploadratio'))
    self.cbhtime.SetValue(Read('uploadtimeh'))
    self.cbmtime.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.cbhtime.GetValue())
    self.utility.config.Write('uploadtimem', self.cbmtime.GetValue())
    self.utility.config.Write('uploadratio', self.cbratio.GetValue())

#####
#
# Class: ColorPanel
#
# Містить опції, які міняють кольор для різних торрентів
#
#####
class ColorPanel(CH_torrent_client_optionPanel):
    def __init__(self, parent, dialog):
        CH_torrent_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 = {}

        # Смугастий 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_torrent_clientTorrentTemp in self.utility.torrents["all"]:
                CH_torrent_clientTorrentTemp.updateColor()

#####
#
# Class: AdvancedNetworkPanel
#
# Містить різні мережні встановлення (типові значення мають бути оптимальні для
більшості користувачів)
#
#####
class AdvancedNetworkPanel(CH_torrent_client_optionPanel):
    def __init__(self, parent, dialog):
        CH_torrent_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_torrent_client_optionPanel):
    def __init__(self, parent, dialog):
        CH_torrent_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.GridSizer(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.allocate_data = wx.SpinCtrl(self, size = wx.Size(60, -1))
        self.allocate_data.SetRange(1, 100)
        datasizer.Add(wx.StaticText(self, -1,
self.utility.lang.get('allocate')), 1, wx.ALIGN_CENTER_VERTICAL)

        allocate_box = wx.BoxSizer(wx.HORIZONTAL)
        allocate_box.Add(self.allocate_data)
        allocate_box.Add(wx.StaticText(self, -1, " " +
self.utility.lang.get('mb') + "/" + self.utility.lang.get("1_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_torrent_clientTree
#
# Розмежованне внесення до всіх панелей опцій
#
#####
class CH_torrent_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('seedingoptions'))

```

```

        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('torrentcolors'))

        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 перемкнутися 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_torrent_client_optionDialog
#
# Створює діалог, який дозволяє користувачам встановлювати різні переваги
#
#####
class CH_torrent_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_torrent_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_torrent_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_torrent_clientParams() #прапор перезапису

    return True

def onOK(self, event = None):
    if self.onApply():
        self.closing = True
        self.saveWindowSettings()

        self.EndModal(wx.ID_OK)

```