BitTorrent Client

GitHub link: https://github.com/GhermanCristian/BitTorrentClient

ACM Classification:

Computer systems organization -> Architectures -> Distributed Architectures -> **Peer-to-peer architectures**

Networks -> Network properties -> **Network privacy and anonymity**

Networks -> Network algorithms -> Control path algorithms -> **Network resources allocation**

Security and privacy -> Security services -> **Digital rights management**

Security and privacy -> Systems security -> **Distributed systems security**

Computing methodologies -> Distributed computing methodologies -> **Distributed algorithms**

AMS Classification:

68M14 – **Distributed systems**

68W15 – **Distributed algorithms**

Programming languages. Frameworks

The program will be predominantly written in **Python 3**. In addition to being easy to use, debug and test, it is also one of the languages used by Bram Cohen, the creator of this protocol, to develop the original BitTorrent client.

Another potential alternative was using **node.js**. However, due to my lack of experience using it, I chose not to use it in this project. Moreover, I believe that **Python** is more suitable for the chosen application architecture (single-module, not client-server).

Some of the more important libraries and frameworks that will be used are:

- **bencode3** - used for parsing bencoded content, such as .torrent files and the tracker response
- **hashlib** - a native **Python** library used in order to encrypt / decrypt some checksum values specific to torrents
- **requests** - used to make HTTP requests
- **Twisted** – framework used for handling simultaneous connections to multiple other users. An alternative to this would have been to create my own event-driven programming loop: however, I chose an existing library because it has already been properly tested and optimized.

The application will have a GUI, which will be implemented in the **tkinter** library. I consider this library to be fairly easy to use, not very wasteful with resources, and able to fulfill the necessary visual and interaction requirements of the program. The **pygame** library was also a good option, and the same could be said for **pyqt**.

Related work

As previously mentioned, the original BitTorrent client, BitTorrent, was also mostly written in **Python**. As with most such clients, it offered features like downloading and uploading multiple files (in parallel), pausing and restarting the download process of a file, or displaying various information such as the completion rate, download speed and ETA. These functionalities will also be present in the current program.

Another popular client, µTorrent, was written in **C++** using the Win32 API. In its early stages it was considered very lightweight; however, nowadays it contains lots of new features which some consider unnecessary. Some of these features include streaming, anti-virus protection, embedded search engines – none of which will feature in this program.

An important difference between several programs available on the market and the current one is their dependency on **advertising** for revenue. This client is not created for-profit and therefore does not need to display advertisements, which use valuable bandwidth and resources and can also be malicious toward users.

Deployment

The application will be ready-to-use (as an **.exe** file) and will not require any installation process. It will be published on a platform such as sourceforge.net, from where the latest builds will be available for download.

References

1. Allen, Kim. *How to make your own bittorrent client*. https://allenkim67.github.io/programming/2016/05/04/how-to-make-your-own-bittorrent-client.html. 2018. Retrieved November 27, 2021.
2. *bencode3 - pypi*. https://pypi.org/project/bencode3/. Retrieved November 28, 2021.
3. *BitTorrent Protocol Specification v1.0*. https://wiki.theory.org/index.php/BitTorrentSpecification. Retrieved November 27, 2021.
4. Cohen, Bran. *The BitTorrent Protocol Specification*. http://www.bittorrent.org/beps/bep_0003.html. Retrieved December 1, 2021
5. *hashlib - Secure hashes and message digests*. https://docs.python.org/3/library/hashlib.html. Retrieved November 28, 2021.
6. *Requests: HTTP for humans*. https://docs.python-requests.org/en/latest/. Retrieved November 28, 2021.
7. *The basics – Twisted 17.9.0 documentation*. https://docs.twistedmatrix.com/en/twisted-17.9.0/core/howto/basics.html. Retrieved November 28, 2021.
8. *tkinter — Python interface to Tcl/Tk*. https://docs.python.org/3/library/tkinter.html. Retrieved November 28, 2021.

9. Widman, Kristen. *How to write a BitTorrent Client, Part 1*. http://www.kristenwidman.com/blog/33/how-to-write-a-bittorrent-client-part-1/. November 23, 2012. Retrieved November 27, 2021.
10. Widman, Kristen. *How to write a BitTorrent Client, Part 2.* http://www.kristenwidman.com/blog/71/how-to-write-a-bittorrent-client-part-2/. November 27, 2012. Retrieved November 27, 2021.