**File Downloader User Guide**

**1. Environment Setup**

1. Install Python3 by running the following command:

$ sudo brew install python3

1. Install click, a Python package for creating command line application, by running the following command:

$ sudo pip3 install click

1. Save the downloader.py file to a folder. For example:

/path/to/downloader.py

**2. Usage:**

1. Open Terminal, and navigate to the folder of downloader.py

$ cd /path/to/downloader.py

1. Users can download the file with or without multi-threading.

Without Multi-Threading:

$ python3 downloader.py http://www.example.com/test.text

With Multi-Threading:

$ python3 downloader.py <http://www.example.com/test.text> -c nThread

nThread will be an argument of number that represents the number of thread will run simultaneously. nThread is limited to no large than 10 to prevent Too Many Threads Exception

1. If download success, a “Download success!” will show. Otherwise, the error message will show.
2. The file is saved in the default download folder.

**3. Design choice:**

I chose Python to develop this application and used click (a python package) to implement the command line arguments and options. The logic and structure of this application is:

1. Read the URL and number of threads from command line arguments
2. Make request to the URL and check if the file is downloadable.
3. Divide the file into nThread pieces by file size
4. Run nThread of DownloadThread to download each piece simultaneously to the default download folder.
5. Save each thread to a list, and combine each piece after download is fully completed.
6. Catch exceptions during making request and download.

**4. Answers to “Some Questions to think about”**

1. What happens if your connection fails during a download?

The application will give a fail message : “Please check network connection and retry”

1. What is the impact your implementation might have on system resource usage?

If there are too many threads running at the same time, it may cause Too Many Threads Exception. To prevent it, I limited the maximum number of threads to 10

1. How do you make sure the user don’t hurt themselves?

I made a few catch exceptions. For example, the application will check if the URL is downloadable, or if the user have permission to download etc.

1. Can you make downloads faster?

I can increase the limit of nThread to a large number. However, the trade-off is that the application may use too much memory.

1. How can you ensure quality does not degrade?

To prevent exceptions, I added a few try catch during making request and downloading file.

5. Sample test cases:

$ python3 downloader.py <http://www.example.com/test.text> -c 3

$ python3 downloader.py <http://www.example.com/test.text>

$ python3 downloader.py [http://www.example111.com/test.text](http://www.example.com/test.text) -c 5

(The above test case will give an error message in terminal)