My Downloader Documentation

Cristian-Ioan Roman

Alexandru Ioan Cuza University, Computer Science Faculty

Abstract. My downloader is an application written in C that contains two components: a client that will tell the server an url and a depth, and a server that will download the resources found at the specific url, down to the lowest level resource that can be found. If the depth specified by the client is more than 0, the server will try to get all the resources it can find up to the relative path given by the client.

Keywords: server · client · http · tcp

1 Introduction

The main objective of this application is to create an application in C or C++ that is able to download from any url the resource found there and to get any other by looking at different depth levels. If any internet connection problem happens while the download is in process, it must restart the download process from the point it was before the internet connection dropped, once the internet is reestablished.
This works by having two components: a server that will process the downloads and a client that will connect to it and will specify the data needed for the server to start the work.

2 Technologies

In order to secure the arrival of packages between server and client TCP connection oriented was used in this project. Besides, the HTTP protocol is a part of the application so that resources found at web locations can be downloaded. As a measure of treating clients in a fast way, the server is implemented on an architecture combining the multi-threading and multiplexing sockets.

3 Application structure

The server is the main component of this application. Accepting a finite number of 15 clients at the same time, these clients are treated individually in separate threads. Here is where a component called 'client_handler' is used by the server to treat any client. It processes the message coming from the client and initializes a very important table for this application: the URL table. Therefore, the server

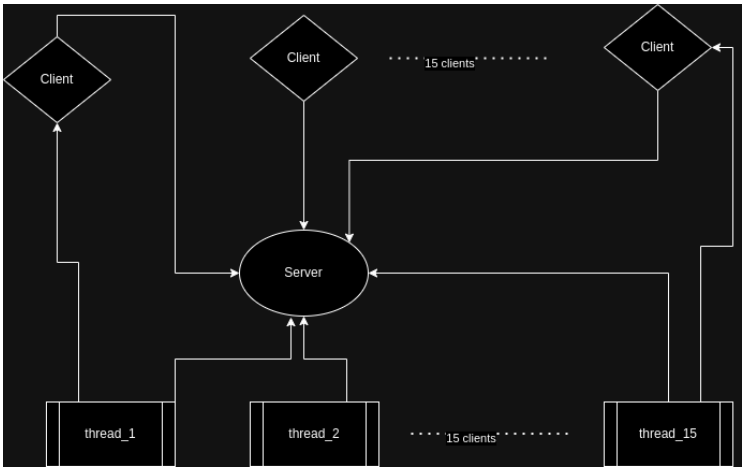

Fig. 1. Server multi-threaded

responds to the client that it has received its request and starts the download of the resources.

This process follows the structure: being at any depth between 0 and the relative depth given by the client + the depth of the URL, it takes every URL from the URL table and downloads them. If it is a page, it will be called "index.html," and a pattern-searching algorithm, Boyer-Moore, will be applied. This will extract all the URLs found in any "index.html" and will add them to the URL table. After finishing the quantity found in the URL table at entry time, it proceeds to the next depth. The downloading function is projected to be recursive so that any depth will be visited twice. This type of architecture ensures that if at any deeper URL is found one at a smaller depth, that URL will be downloaded too.

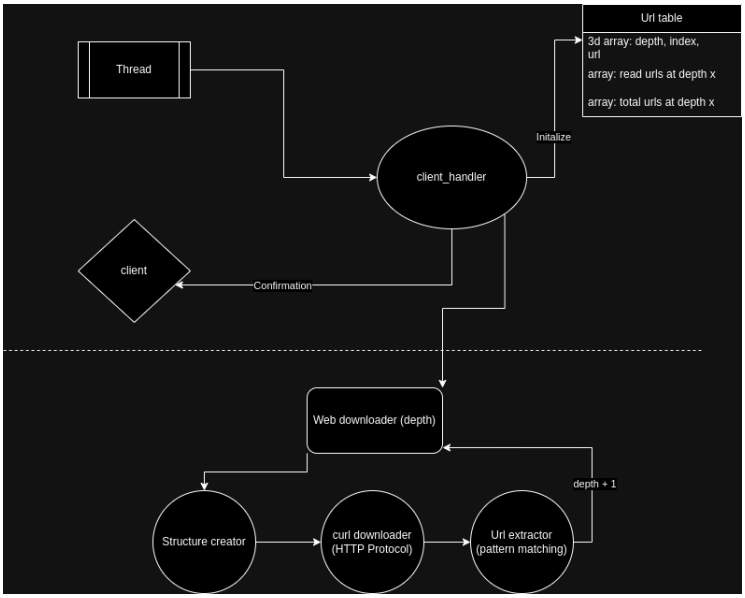

Fig. 2. web handler

4 Implementation aspects

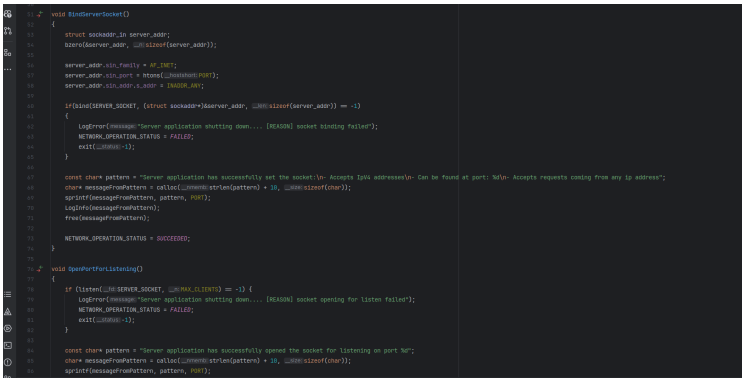

Fig. 3. Server connection settings

Fig. 4. Url table

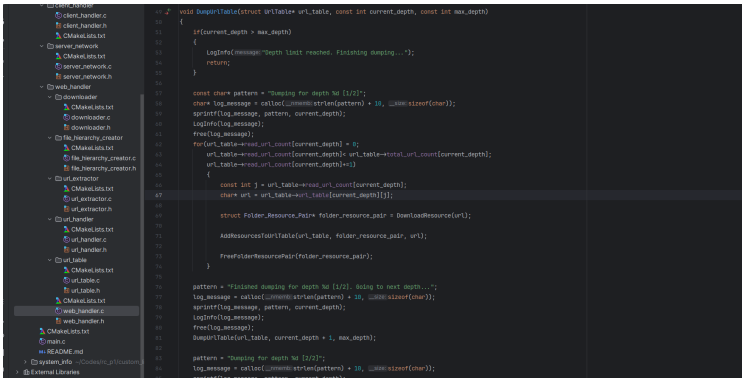

Fig. 5. Web handler

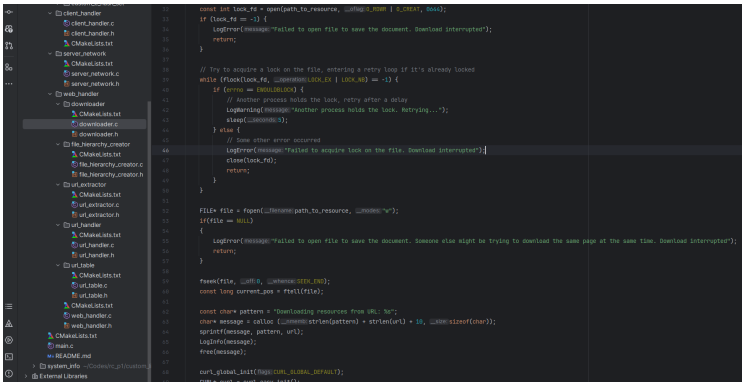

Fig. 6. curl-downloader

Fig. 7. Boyer moore pattern searching

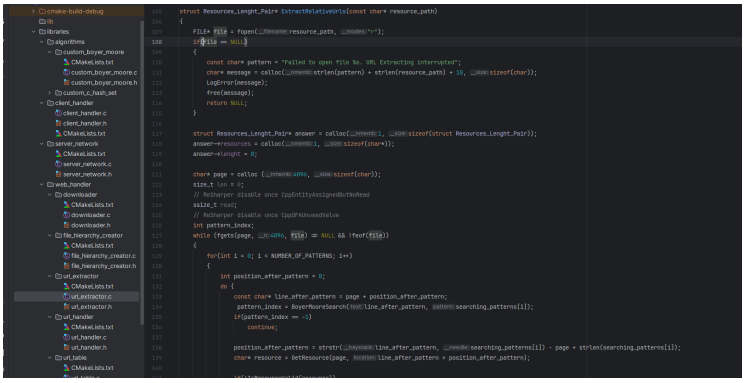

Fig. 8. Url extractor based on boyer moore

5 Future directions

To enhance the versatility and functionality of the application, a natural progression would be to extend its resource downloading capabilities beyond the HTTP protocol. One widely used and standardized protocol for file transfer is the File Transfer Protocol (FTP). Incorporating FTP support into the application can provide users with the flexibility to download resources from various machine locations.

Eventually, the pattern searching algorithm can be improved in order to identify the urls found in other pages faster and with more accuracy. Currently, the server tries to download innexistent resources too.

References

1. Computer networks course page, https://profs.info.uaic.ro/ computernetworks/cursullaboratorul.php. Last accessed 9 Ian 2024
2. Assistant Teacher page, https://www.andreis.ro/teaching/computer-networks. Last accessed 9 Ian 2024
3. Linux man page, https://linux.die.net/man/. Last accessed 26 Dec 2023