

Coding Project 2

CS 6233

The purpose of this exercise is to get more familiar with concurrency and synchronization issues as covered in the last two lectures. As a result, there are two parts to this assignment.

Instructions

There is a zip file to go along with this assignment in the Coding Projects 1 directory called Source and Data Files. This zip file contains a C source code file called “fileparse.c” and a directory with a series of Apache log files called “access_logs”.

If you are not familiar with how to compile the file (and you should be), perform the following on a linux machine.

```
gcc -o fileparse fileparse.c
./fileparse <directory>
```

When turning the project in, make sure you have commented your code properly and that it compiles. If the code doesn't compile, I won't accept it.

Part 1

This program is currently a single threaded process. Your objective is to make this a multi-threaded program such that the reading of files is done concurrently.

To do this, you'll need to use the pthreads library as discussed in class. Be careful to ensure that all threads have completed running before exiting and that you have destroyed any objects you created.

Part 2

The current fileparse.c loops through logs and displays the line and the size of the line. The objective of the second part of this exercise is to get a count of the unique number of IP addresses in the log file. Note: the IP address is the first set of numbers on each line and it indicates the IP of the person that visited that page.

You can take any approach (in C – no calls to outside programs) you would like to do this but keep in mind that it must be free of race conditions. I would recommend using synchronization primitives provided by the pthreads library but other techniques such as using thread local storage would also be acceptable.

Extra materials:

pthreads Tutorial (<https://computing.llnl.gov/tutorials/pthreads/>)