Jiashu Liao (jl1811), Chijun Sha (cs1143)
Steve Hsu (slh249), Boyang Fu (bf249)

of Rutgers

Project 3 Analysis

CS 214

Systems Programming

Submitted to:
Instructor Andrew Tjang
Ananya Jana
Zhe Wang
Gang Qiao
Hanxiong Chen

Contents

1 Analysis

Project design:
Server:
We used a linked list to control the data flows. To be more specific, we use each node to store the information of a particular client for later send back. We assign a session ID to each client. Then, the client interacts with server with its unique session ID. Therefore we can identify which client sent the data. For each socket, we create a thread to serve it. If a thread receives a "DUMP" (QuitServer) request, it will check if COUNTER (a counter to identify whether all buffer is stored correctly for the client) is equal to 0, which servers as a way to detect whether all the files are stored into a matrix before printing the matrix out. Then we send the sorted a buffer containing all the sorted data back to client.
Client:
The client will check all files in the input directory and its subdirectories. It will send all CSV files to the server. It will send the "DUMP" (QuitServer) request after sending all CSV files. Then it will wait until it gets a "Sorting Completed" signal from the server. Then it will get the All-Sorted file back.
How to use our code:

```
make
./sorter_server -p 12345
./sorter_client -c food -h grep.cs.rutgers.edu -p 12345 -d thisdir/thatdir
-o anotherdir
```

2 Extra Credit

Extra 1:
So basically I designed this first extra credit by using a semaphore for the client. What does that mean? Since the server is design to get every connect and in the client side to input the pool (number). If you don't input pool, there will not be any limit for the numbers of socket, however, if you input a number for pool, for this specific client it will have most (this number) sockets to connect and others will wait by semaphore.
Extra 2:
We allocate a linked list to store all session IDs to identify the owner/origins of the connected socket. The server will give each client a unique session ID. All clients will ask the server to get an ID at first, then interact with server using the session ID.