CSCE PA1 – Report

Video Demo - https://youtu.be/vfajNhY8DYE

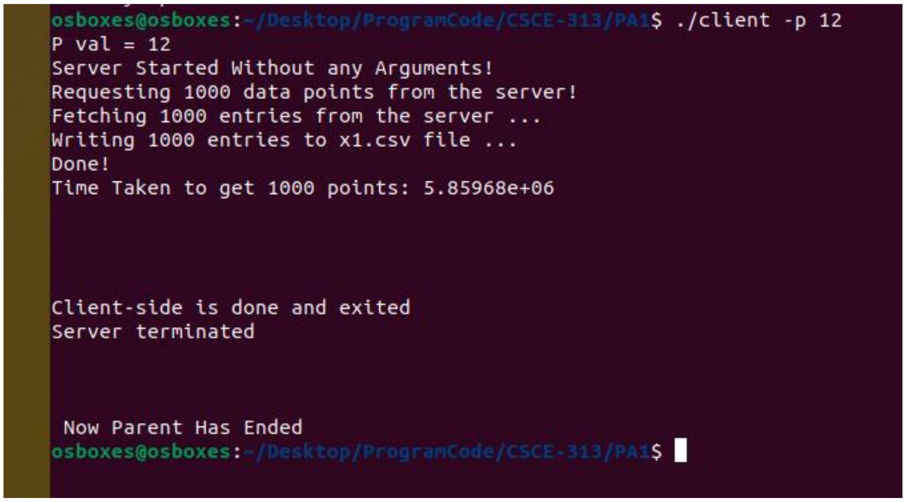
Task #1. Collecting 1000 data points and getting the result:

Time Taken to get 1000 points: 5.85968e+06 for person 12

Time Taken to get 1000 points: 6.02761e+06 for person 12

Time Taken to get 1000 points: 6.00130e+06 for person 12

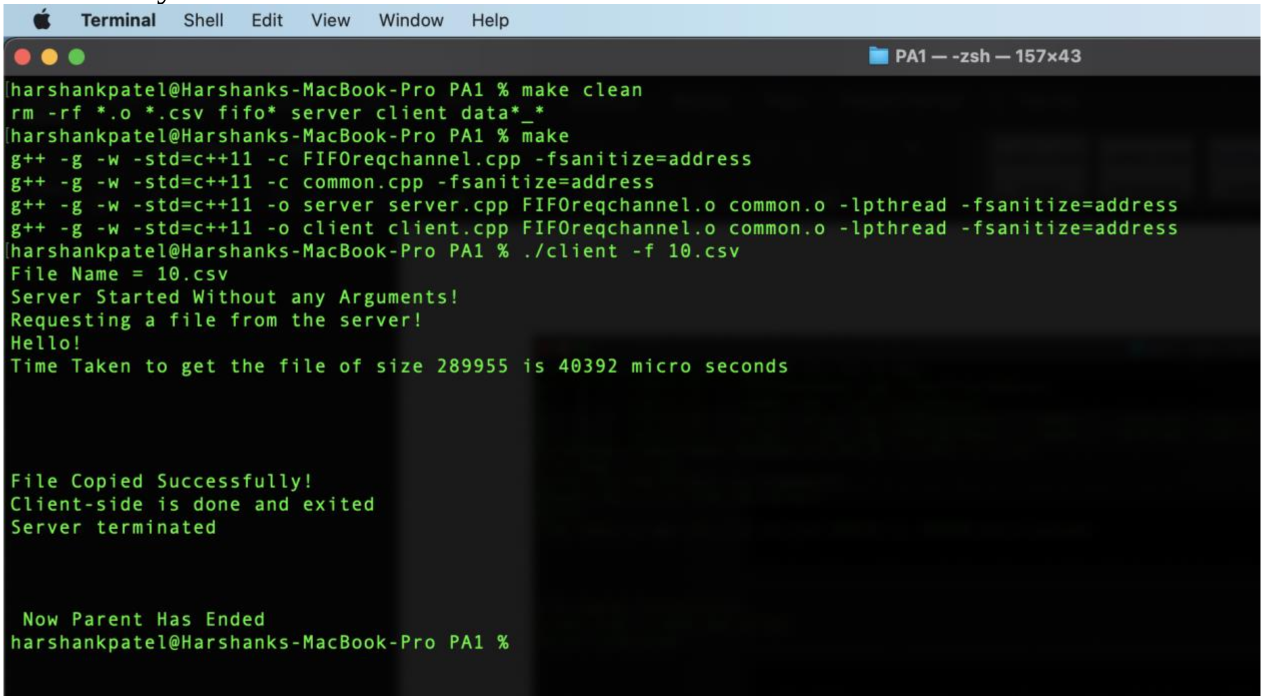
Time Taken to get 1000 points: 5.99491e+06 for person 12



```
osboxes@osboxes:~/Desktop/ProgramCode/CSCE-313/PA1$ ./client -p 12
P val = 12
Server Started Without any Arguments!
Requesting 1000 data points from the server!
Fetching 1000 entries from the server ...
Writing 1000 entries to x1.csv file ...
Done!
Time Taken to get 1000 points: 5.85968e+06




Client-side is done and exited
Server terminated



 Now Parent Has Ended
osboxes@osboxes:~/Desktop/ProgramCode/CSCE-313/PA1$ 
```

Task: Check for memory Leak!

Proof: I used -fsanitize=address in my make file to compile and ran the program, and no memory leak error was thrown. This means no MEMORY LEAK!

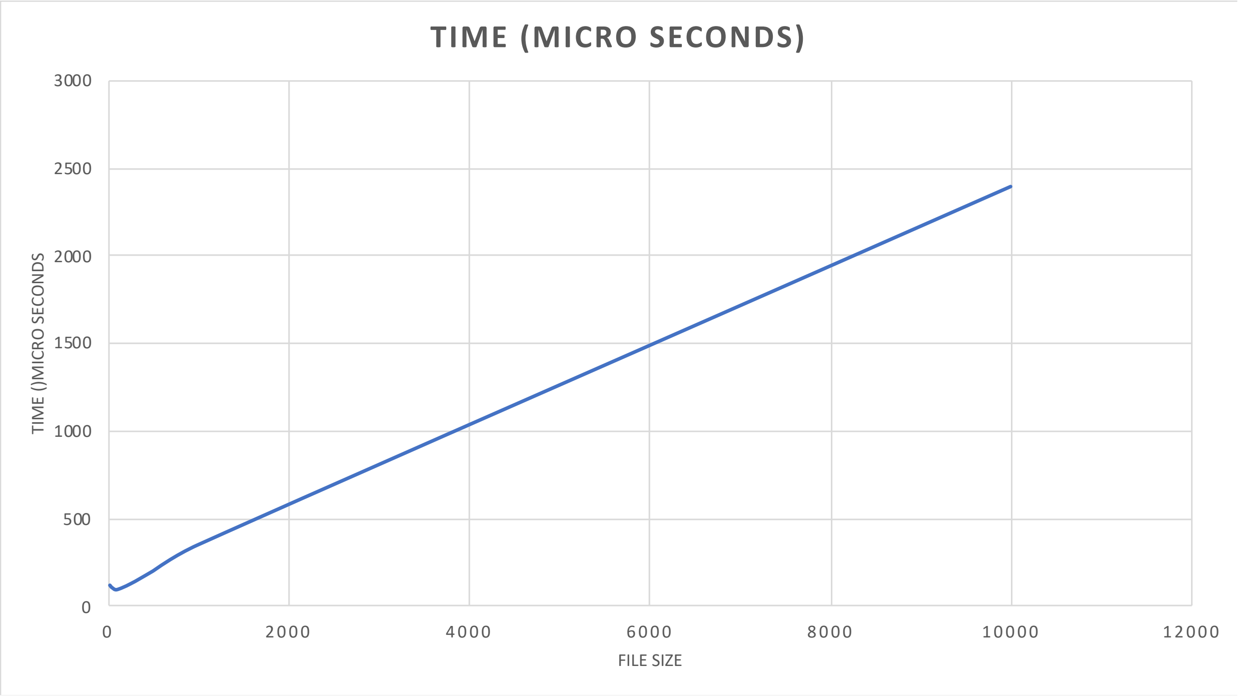


```
[harshankpatel@Harshanks-MacBook-Pro PA1 % make clean
rm -rf *.o *.csv fifo* server client data*_*
[harshankpatel@Harshanks-MacBook-Pro PA1 % make
g++ -g -w -std=c++11 -c FIFOreqchannel.cpp -fsanitize=address
g++ -g -w -std=c++11 -c common.cpp -fsanitize=address
g++ -g -w -std=c++11 -o server server.cpp FIFOreqchannel.o common.o -lpthread -fsanitize=address
g++ -g -w -std=c++11 -o client client.cpp FIFOreqchannel.o common.o -lpthread -fsanitize=address
[harshankpatel@Harshanks-MacBook-Pro PA1 % ./client -f 10.csv
File Name = 10.csv
Server Started Without any Arguments!
Requesting a file from the server!
Hello!
Time Taken to get the file of size 289955 is 40392 micro seconds




File Copied Successfully!
Client-side is done and exited
Server terminated




 Now Parent Has Ended
harshankpatel@Harshanks-MacBook-Pro PA1 %
```

Task #2.

Size (Bytes)	**Time (micro seconds)**	**Buffer Size**
25	**115**	**256**
50	**101**	**256**
100	**90**	**256**
256	**124**	**256**
500	**196**	**256**
1000	**347**	**256**
10000	**2397**	**256**
100000	**11863**	**256**

Graph of Time taken in microseconds vs the size of the file downloaded



Task 2: Varying Buffer Size

Size (Bytes)	Time (micro seconds)	Buffer Size
10000	9726	50
10000	3417	100
10000	2362	200
10000	1667	500
10000	983	700
10000	582	800
10000	811	900
10000	882	1000
10000	156	10000

As you can see the more the buffer the less the runtime is for a large file. The reason is due to the larger the buffer, the more the fille can copy in one pass, the faster your program is.
The Buffer is responsible for storing some data and then transferring this to or from the Hard Disk. This results in increased speed and more efficient transfer between the computer and the Hard Disk. This also reduces the movement of the read/write head across the magnetic disk which results in reduced wear and tear.

Task: An explanation for the various command-line commands you could type when running the client:

Ans: The various command line commands are mostly listed here:

Make: Runs the compilation of the cpp and header files.

Make clean: cleans the .o files

./client – Runs the program with extracting one specific data point

./client -c – Requests a new channel and performs 5 data points extraction.

./client -m 500 -f 1.csv – sets the buffer size to 500 and then extracts/downloads 1.csv file to the received folder.

./client -m 500 – Runs the collection of 1 data point from server with the buffer size of 500

./client -p 10 – Gets 1K values of patient 10

./client -p 1 -t 0.00 -e 2 – Gets patient 1's ecg2 value at time stamp 0.00