COMP 8005 Assignment 2 Testing
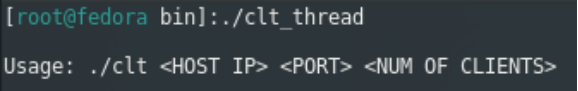
Alex Zielinski – A00803488
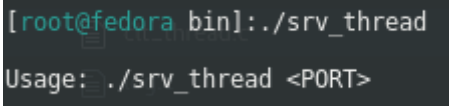
Contents

Testing Explained

Two machines were used for testing. The machine that was used to run the three server programs has the IP **192.168.0.18**. The machine that was used to run the client program has the IP **192.168.0.20**. The server machine listened on port 7000. The client sent packets of size 1000 bytes to the server. The server would then simply echo back the received packet back to the client. The client sent packets to the server none stop for a duration of 20 seconds for each test. The client program also created 2000 worker clients to simulate multiple connections. So for each test the server had to accommodate 2000 client connections, each sending packets of size 1000 bytes for 20 seconds.
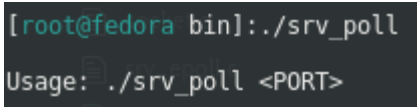
Note: whenever the bolded words **IP, PORT** and **NUM** are encountered this simply signifies that if a different individual was to do these tests they would enter an IP, port and number of clients specific to them. For this test **IP** refers to **192.168.0.18**, **PORT** refers to **7000** and **NUM** refers to **2000**.
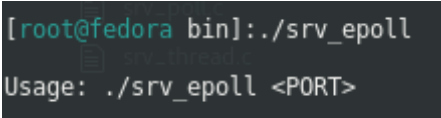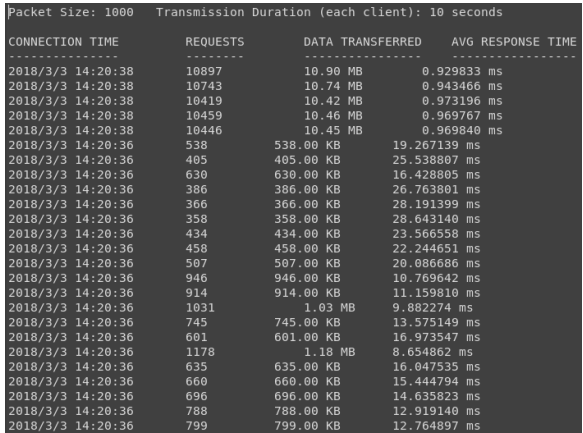
Test Cases

Program Usage Tests

Test Case #1	Expected	Screenshot(s)	Result
Steps: 1. Run the client program as follows **./clt_thread** 2. Notice the terminal output	Error will pop up indicate the programs usage.	`[root@fedora bin]:./clt_thread` `Usage: ./clt <HOST IP> <PORT> <NUM OF CLIENTS>`	**PASS**
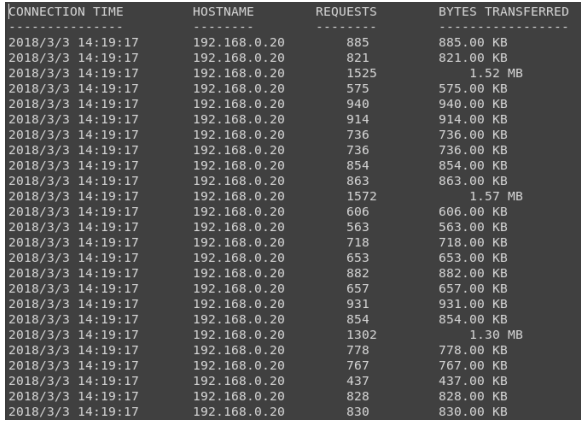
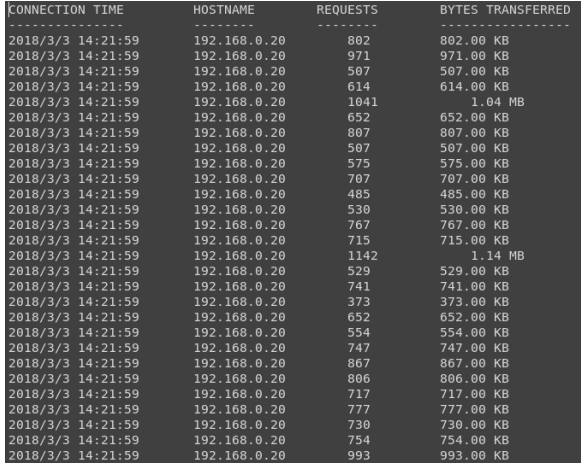| Test Case #2	Expected	Screenshot(s)	Result
Steps: 1. Run the multi-threaded server program as follows **./srv_thread** 1. Notice the terminal output	Error will pop up indicate the programs usage.	[root@fedora bin]:./srv_thread Usage: ./srv_thread <PORT>	**PASS**

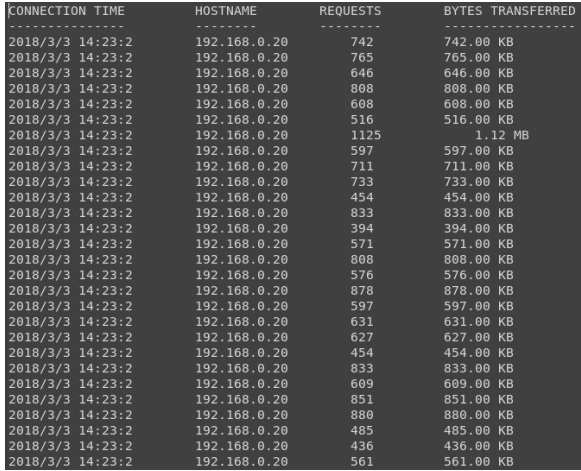| Test Case #3	Expected	Screenshot(s)	Result
Steps: 1. Run the multi-plexed server program as follows **./srv_poll** 2. Notice the terminal output	Error will pop up indicate the programs usage.	[root@fedora bin]:./srv_poll Usage: ./srv_poll <PORT>	**PASS**

Test Case #4	Expected	Screenshot(s)	Result
Steps: 1. Run the asynch-ronous server program as follows **./srv_epoll** 3. Notice the terminal output	Error will pop up indicate the programs usage.	[root@fedora bin]:./srv_epoll Usage: ./srv_epoll <PORT>	**PASS**

Log Entry Tests

Test Case #5	Expected	Screenshot(s)	Result
| **Steps:**<br>1. Run the multithreaded server on a machine<br><br>**./srv_thread IP PORT**<br><br>2. Run the client program on a different machine<br><br>**./clt_thread PORT NUM**<br><br>3. Wait for the client program to shut down<br><br>4. On the server press **ctrl + c** to shut it down<br><br>5. On the client navigate to the **/data** folder and open the file called **clt_log**<br><br>6. Notice the logged data | The log file should contain data regarding the time of connection, the number of requests processed, and the amount of data sent (in bytes). At the top of the log file there should be a line explaining the test case where is declares the packet size and the transmission duration. | ```
Packet Size: 1000    Transmission Duration (each client): 10 seconds

CONNECTION TIME       REQUESTS        DATA TRANSFERRED     AVG RESPONSE TIME
---------------       --------        ----------------     -----------------
2018/3/3 14:20:38     10897           10.90 MB             0.929833 ms
2018/3/3 14:20:38     10743           10.74 MB             0.943466 ms
2018/3/3 14:20:38     10419           10.42 MB             0.973196 ms
2018/3/3 14:20:38     10459           10.46 MB             0.969767 ms
2018/3/3 14:20:38     10446           10.45 MB             0.969840 ms
2018/3/3 14:20:36     538             538.00 KB            19.267139 ms
2018/3/3 14:20:36     405             405.00 KB            25.538807 ms
2018/3/3 14:20:36     630             630.00 KB            16.428805 ms
2018/3/3 14:20:36     386             386.00 KB            26.763801 ms
2018/3/3 14:20:36     366             366.00 KB            28.191399 ms
2018/3/3 14:20:36     358             358.00 KB            28.643140 ms
2018/3/3 14:20:36     434             434.00 KB            23.566558 ms
2018/3/3 14:20:36     458             458.00 KB            22.244651 ms
2018/3/3 14:20:36     507             507.00 KB            20.086686 ms
2018/3/3 14:20:36     946             946.00 KB            10.769642 ms
2018/3/3 14:20:36     914             914.00 KB            11.159810 ms
2018/3/3 14:20:36     1031            1.03 MB              9.882274 ms
2018/3/3 14:20:36     745             745.00 KB            13.575149 ms
2018/3/3 14:20:36     601             601.00 KB            16.973547 ms
2018/3/3 14:20:36     1178            1.18 MB              8.654862 ms
2018/3/3 14:20:36     635             635.00 KB            16.047535 ms
2018/3/3 14:20:36     660             660.00 KB            15.444794 ms
2018/3/3 14:20:36     696             696.00 KB            14.635823 ms
2018/3/3 14:20:36     788             788.00 KB            12.919140 ms
2018/3/3 14:20:36     799             799.00 KB            12.764897 ms
``` | **PASS** |

Test Case #6	Expected	Screenshot(s)	Result
| **Steps:**<br><br>1. Run the multithreaded server on a machine<br><br>**./srv_thread IP PORT**<br><br>2. Run the client program on a different machine<br><br>**./clt_thread PORT NUM**<br><br>3. Wait for the client program to shut down<br><br>4. On the server press **ctrl + c** to shut it down<br><br>5. On the server navigate to the **/data** folder and open the file called **srv_thread_log**<br><br>6. Notice the logged data | The log file should contain data regarding the time of connection, the IP of the connected client, the number of requests processed, and the amount of data processed (in bytes). | ```
CONNECTION TIME        HOSTNAME        REQUESTS      BYTES TRANSFERRED
---------------        --------        --------      -----------------
2018/3/3 14:19:17      192.168.0.20    885           885.00 KB
2018/3/3 14:19:17      192.168.0.20    821           821.00 KB
2018/3/3 14:19:17      192.168.0.20    1525              1.52 MB
2018/3/3 14:19:17      192.168.0.20    575           575.00 KB
2018/3/3 14:19:17      192.168.0.20    940           940.00 KB
2018/3/3 14:19:17      192.168.0.20    914           914.00 KB
2018/3/3 14:19:17      192.168.0.20    736           736.00 KB
2018/3/3 14:19:17      192.168.0.20    736           736.00 KB
2018/3/3 14:19:17      192.168.0.20    854           854.00 KB
2018/3/3 14:19:17      192.168.0.20    863           863.00 KB
2018/3/3 14:19:17      192.168.0.20    1572              1.57 MB
2018/3/3 14:19:17      192.168.0.20    606           606.00 KB
2018/3/3 14:19:17      192.168.0.20    563           563.00 KB
2018/3/3 14:19:17      192.168.0.20    718           718.00 KB
2018/3/3 14:19:17      192.168.0.20    653           653.00 KB
2018/3/3 14:19:17      192.168.0.20    882           882.00 KB
2018/3/3 14:19:17      192.168.0.20    657           657.00 KB
2018/3/3 14:19:17      192.168.0.20    931           931.00 KB
2018/3/3 14:19:17      192.168.0.20    854           854.00 KB
2018/3/3 14:19:17      192.168.0.20    1302              1.30 MB
2018/3/3 14:19:17      192.168.0.20    778           778.00 KB
2018/3/3 14:19:17      192.168.0.20    767           767.00 KB
2018/3/3 14:19:17      192.168.0.20    437           437.00 KB
2018/3/3 14:19:17      192.168.0.20    828           828.00 KB
2018/3/3 14:19:17      192.168.0.20    830           830.00 KB
``` | **PASS** |
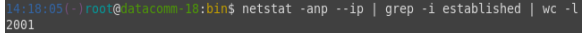
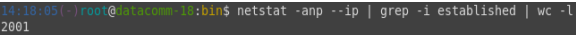| Test Case #7	Expected	Screenshot(s)	Result
Steps: 1. Run the multiplexed server on a machine **./srv_poll IP PORT** 2. Run the client program on a different machine **./clt_thread PORT NUM** 3. Wait for the client program to shut down 4. Wait for the server program to shut down 5. On the server navigate to the **/data** folder and open the file called **srv_poll_log** 6. Notice the logged data	The log file should contain data regarding the time of connection, the ip of the connected client, the number of requests processed, and the amount of data processed (in bytes).	<table><tr><td>CONNECTION TIME</td><td>HOSTNAME</td><td>REQUESTS</td><td>BYTES TRANSFERRED</td></tr><tr><td>2018/3/3 14:21:59</td><td>192.168.0.20</td><td>802</td><td>802.00 KB</td></tr><tr><td>2018/3/3 14:21:59</td><td>192.168.0.20</td><td>971</td><td>971.00 KB</td></tr><tr><td>2018/3/3 14:21:59</td><td>192.168.0.20</td><td>507</td><td>507.00 KB</td></tr><tr><td>2018/3/3 14:21:59</td><td>192.168.0.20</td><td>614</td><td>614.00 KB</td></tr><tr><td>2018/3/3 14:21:59</td><td>192.168.0.20</td><td>1041</td><td>1.04 MB</td></tr><tr><td>2018/3/3 14:21:59</td><td>192.168.0.20</td><td>652</td><td>652.00 KB</td></tr><tr><td>2018/3/3 14:21:59</td><td>192.168.0.20</td><td>807</td><td>807.00 KB</td></tr><tr><td>2018/3/3 14:21:59</td><td>192.168.0.20</td><td>507</td><td>507.00 KB</td></tr><tr><td>2018/3/3 14:21:59</td><td>192.168.0.20</td><td>575</td><td>575.00 KB</td></tr><tr><td>2018/3/3 14:21:59</td><td>192.168.0.20</td><td>707</td><td>707.00 KB</td></tr><tr><td>2018/3/3 14:21:59</td><td>192.168.0.20</td><td>485</td><td>485.00 KB</td></tr><tr><td>2018/3/3 14:21:59</td><td>192.168.0.20</td><td>530</td><td>530.00 KB</td></tr><tr><td>2018/3/3 14:21:59</td><td>192.168.0.20</td><td>767</td><td>767.00 KB</td></tr><tr><td>2018/3/3 14:21:59</td><td>192.168.0.20</td><td>715</td><td>715.00 KB</td></tr><tr><td>2018/3/3 14:21:59</td><td>192.168.0.20</td><td>1142</td><td>1.14 MB</td></tr><tr><td>2018/3/3 14:21:59</td><td>192.168.0.20</td><td>529</td><td>529.00 KB</td></tr><tr><td>2018/3/3 14:21:59</td><td>192.168.0.20</td><td>741</td><td>741.00 KB</td></tr><tr><td>2018/3/3 14:21:59</td><td>192.168.0.20</td><td>373</td><td>373.00 KB</td></tr><tr><td>2018/3/3 14:21:59</td><td>192.168.0.20</td><td>652</td><td>652.00 KB</td></tr><tr><td>2018/3/3 14:21:59</td><td>192.168.0.20</td><td>554</td><td>554.00 KB</td></tr><tr><td>2018/3/3 14:21:59</td><td>192.168.0.20</td><td>747</td><td>747.00 KB</td></tr><tr><td>2018/3/3 14:21:59</td><td>192.168.0.20</td><td>867</td><td>867.00 KB</td></tr><tr><td>2018/3/3 14:21:59</td><td>192.168.0.20</td><td>806</td><td>806.00 KB</td></tr><tr><td>2018/3/3 14:21:59</td><td>192.168.0.20</td><td>717</td><td>717.00 KB</td></tr><tr><td>2018/3/3 14:21:59</td><td>192.168.0.20</td><td>777</td><td>777.00 KB</td></tr><tr><td>2018/3/3 14:21:59</td><td>192.168.0.20</td><td>730</td><td>730.00 KB</td></tr><tr><td>2018/3/3 14:21:59</td><td>192.168.0.20</td><td>754</td><td>754.00 KB</td></tr><tr><td>2018/3/3 14:21:59</td><td>192.168.0.20</td><td>993</td><td>993.00 KB</td></tr></table>	**PASS**

Test Case #8	Expected	Screenshot(s)	Result
Steps: 1. Run the asynchronous server on a machine *./srv_epoll IP PORT* 2. Run the client program on a different machine *./clt_thread PORT NUM* 3. Wait for the client program to shut down 4. Wait for the server program to shut down 5. On the server navigate to the **/data** folder and open the file called **srv_epoll_log** 6. Notice the logged data	The log file should contain data regarding the time of connection, the ip of the connected client, the number of requests processed, and the amount of data processed (in bytes).	<table><tr><td>CONNECTION TIME</td><td>HOSTNAME</td><td>REQUESTS</td><td>BYTES TRANSFERRED</td></tr><tr><td>2018/3/3 14:23:2</td><td>192.168.0.20</td><td>742</td><td>742.00 KB</td></tr><tr><td>2018/3/3 14:23:2</td><td>192.168.0.20</td><td>765</td><td>765.00 KB</td></tr><tr><td>2018/3/3 14:23:2</td><td>192.168.0.20</td><td>646</td><td>646.00 KB</td></tr><tr><td>2018/3/3 14:23:2</td><td>192.168.0.20</td><td>808</td><td>808.00 KB</td></tr><tr><td>2018/3/3 14:23:2</td><td>192.168.0.20</td><td>608</td><td>608.00 KB</td></tr><tr><td>2018/3/3 14:23:2</td><td>192.168.0.20</td><td>516</td><td>516.00 KB</td></tr><tr><td>2018/3/3 14:23:2</td><td>192.168.0.20</td><td>1125</td><td>1.12 MB</td></tr><tr><td>2018/3/3 14:23:2</td><td>192.168.0.20</td><td>597</td><td>597.00 KB</td></tr><tr><td>2018/3/3 14:23:2</td><td>192.168.0.20</td><td>711</td><td>711.00 KB</td></tr><tr><td>2018/3/3 14:23:2</td><td>192.168.0.20</td><td>733</td><td>733.00 KB</td></tr><tr><td>2018/3/3 14:23:2</td><td>192.168.0.20</td><td>454</td><td>454.00 KB</td></tr><tr><td>2018/3/3 14:23:2</td><td>192.168.0.20</td><td>833</td><td>833.00 KB</td></tr><tr><td>2018/3/3 14:23:2</td><td>192.168.0.20</td><td>394</td><td>394.00 KB</td></tr><tr><td>2018/3/3 14:23:2</td><td>192.168.0.20</td><td>571</td><td>571.00 KB</td></tr><tr><td>2018/3/3 14:23:2</td><td>192.168.0.20</td><td>808</td><td>808.00 KB</td></tr><tr><td>2018/3/3 14:23:2</td><td>192.168.0.20</td><td>576</td><td>576.00 KB</td></tr><tr><td>2018/3/3 14:23:2</td><td>192.168.0.20</td><td>878</td><td>878.00 KB</td></tr><tr><td>2018/3/3 14:23:2</td><td>192.168.0.20</td><td>597</td><td>597.00 KB</td></tr><tr><td>2018/3/3 14:23:2</td><td>192.168.0.20</td><td>631</td><td>631.00 KB</td></tr><tr><td>2018/3/3 14:23:2</td><td>192.168.0.20</td><td>627</td><td>627.00 KB</td></tr><tr><td>2018/3/3 14:23:2</td><td>192.168.0.20</td><td>454</td><td>454.00 KB</td></tr><tr><td>2018/3/3 14:23:2</td><td>192.168.0.20</td><td>833</td><td>833.00 KB</td></tr><tr><td>2018/3/3 14:23:2</td><td>192.168.0.20</td><td>609</td><td>609.00 KB</td></tr><tr><td>2018/3/3 14:23:2</td><td>192.168.0.20</td><td>851</td><td>851.00 KB</td></tr><tr><td>2018/3/3 14:23:2</td><td>192.168.0.20</td><td>880</td><td>880.00 KB</td></tr><tr><td>2018/3/3 14:23:2</td><td>192.168.0.20</td><td>485</td><td>485.00 KB</td></tr><tr><td>2018/3/3 14:23:2</td><td>192.168.0.20</td><td>436</td><td>436.00 KB</td></tr><tr><td>2018/3/3 14:23:2</td><td>192.168.0.20</td><td>561</td><td>561.00 KB</td></tr></table>	**PASS**

Netstat Tests

Test Case #9	Expected	Screenshot(s)	Result
Steps: 1. Run the multithreaded server on a machine **./srv_thread IP PORT** 2. Run the client program on a different machine **./clt_thread PORT NUM** 3. On the server machine, open a terminal and keep entering the following command until the output reaches 2001 **netstat -anp –ip | grep -i established | wc -l**	The output of the netstat command should say **2001**. This indicates that there are 2000 clients connected to the server and 1 listening socket.	`14:18:05(~)root@datacomm-18:bin$ netstat -anp --ip | grep -i established | wc -l` `2001`	**PASS**

Test Case #10	Expected	Screenshot(s)	Result
Steps: 1. Run the multiplexed server on a machine **./srv_poll IP PORT** 2. Run the client program on a different machine **./clt_thread PORT NUM** 3. On the server machine, open a terminal and keep entering the following command until the output reaches 2001 **netstat -anp –ip | grep -i established | wc -l**	The output of the netstat command should say *2001*. This indicates that there are 2000 clients connected to the server and 1 listening socket.	`14:18:05(-)root@datacomm-18:bin$ netstat -anp --ip	grep -i established	wc -l` `2001`	**PASS**

Test Case #11	Expected	Screenshot(s)	Result
Steps: 1. Run the multiplexed server on a machine **./srv_poll IP PORT** 2. Run the client program on a different machine **./clt_thread PORT NUM** 3. On the server machine, open a terminal and keep entering the following command until the output reaches 2001 **netstat -anp –ip | grep -i established | wc -l**	The output of the netstat command should say ***2001***. This indicates that there are 2000 clients connected to the server and 1 listening socket.	`14:18:05(~)root@datacomm-18:bin$ netstat -anp --ip | grep -i established | wc -l` `2001`	**PASS**

Log Entry Number of Connection Tests

Test Case #12	Expected	Screenshot(s)	Result
| **Steps:**<br>1. Run the multithreaded server on a machine<br><br>**./srv_thread IP PORT**<br><br>2. Run the client program on a different machine<br><br>**./clt_thread PORT NUM**<br><br>3. Wait for the client program to shut down<br><br>4. On the server press **ctrl + c** to shut it down<br><br>5. On the server navigate to the **/data** folder and open the file called **srv_thread_log** and scroll to the bottom of the log file | The bottom of the log file should contain the number of clients that connected to the server which in this case would be 2000. | ```
--------------------------------
Total Client Connections: 2000
``` | **PASS** |

Test Case #13	Expected	Screenshot(s)	Result
Steps: 1. Run the multithreaded server on a machine **./srv_poll IP PORT** 2. Run the client program on a different machine **./clt_thread PORT NUM** 3. Wait for the client program to shut down 4. On the server press **ctrl + c** to shut it down 5. On the server navigate to the **/data** folder and open the file called **srv_poll_log** and scroll to the bottom of the log file	The bottom of the log file should contain the number of clients that connected to the server which in this case would be 2000.	-- Total Client Connections: 2000	**PASS**

Test Case #14	Expected	Screenshot(s)	Result
Steps: 1. Run the multithreaded server on a machine **./srv_epoll IP PORT** 2. Run the client program on a different machine **./clt_thread PORT NUM** 3. Wait for the client program to shut down 4. On the server press **ctrl + c** to shut it down 5. On the server navigate to the **/data** folder and open the file called **srv_epoll_log** and scroll to the bottom of the log file	The bottom of the log file should contain the number of clients that connected to the server which in this case would be 2000.	------------------------------------ Total Client Connections: 2000	**PASS**