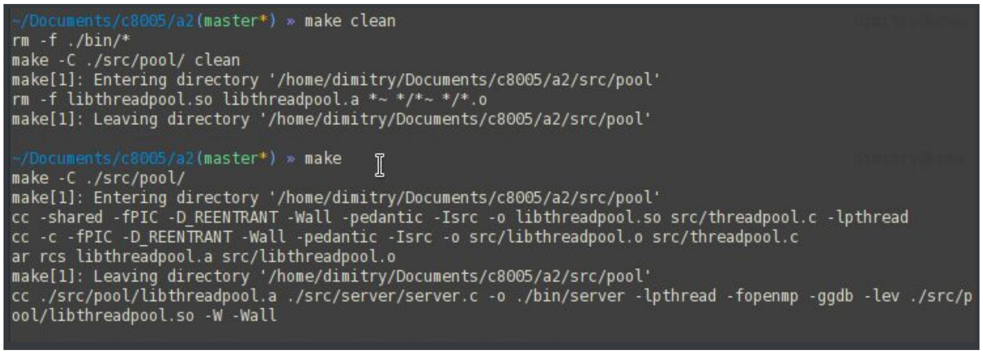
COMP 8005 A2 Testing

<table>
<tr><td>Test#</td><td>1 &amp; 2</td></tr>
<tr><td>Description</td><td>Clear old files and compile new executables.</td></tr>
<tr><td>Steps</td><td>

- Run make clean to clear
  - bin/
  - src/pool/
  - src/pool/src/
- Run make to compile
  - bin/server
  - src/pool/libthreadpool.so
  - src/pool/libthreadpool.a

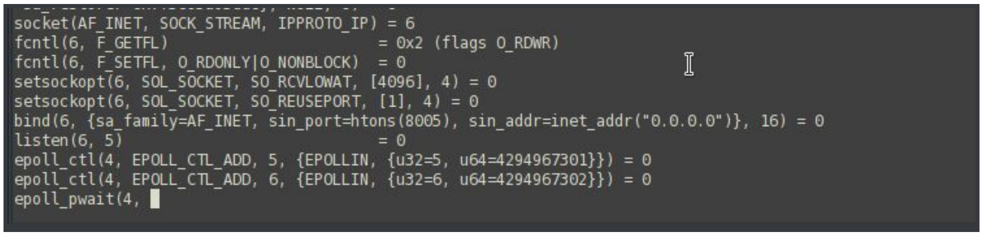
</td></tr>
<tr><td>Result</td><td>



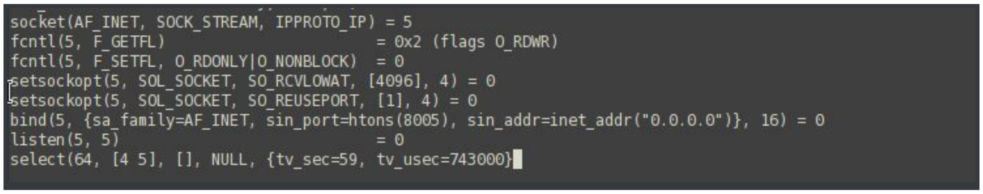
```
~/Documents/c8005/a2(master*) » make clean
rm -f ./bin/*
make -C ./src/pool/ clean
make[1]: Entering directory '/home/dimitry/Documents/c8005/a2/src/pool'
rm -f libthreadpool.so libthreadpool.a *~ */*~ */*.o
make[1]: Leaving directory '/home/dimitry/Documents/c8005/a2/src/pool'

~/Documents/c8005/a2(master*) » make
make -C ./src/pool/
make[1]: Entering directory '/home/dimitry/Documents/c8005/a2/src/pool'
cc -shared -fPIC -D_REENTRANT -Wall -pedantic -Isrc -o libthreadpool.so src/threadpool.c -lpthread
cc -c -fPIC -D_REENTRANT -Wall -pedantic -Isrc -o src/libthreadpool.o src/threadpool.c
ar rcs libthreadpool.a src/libthreadpool.o
make[1]: Leaving directory '/home/dimitry/Documents/c8005/a2/src/pool'
cc ./src/pool/libthreadpool.a ./src/server/server.c -o ./bin/server -lpthread -fopenmp -ggdb -lev ./src/p
ool/libthreadpool.so -W -Wall
```

</td></tr>
<tr><td>Success</td><td>Yes</td></tr>
</table>

Test#	3
Description	Ensure that epoll runs and uses non-blocking sockets.
Steps	● strace the server with epoll argument ○ Make sure server blocks on epoll_pwait and not on accept ○ Make sure socket is successfully set to SO_NONBLOCK and SO_REUSEPORT ○ Make sure SO_RCVLOWAT is set to BUFSIZE=4096
Result	
Success	Yes

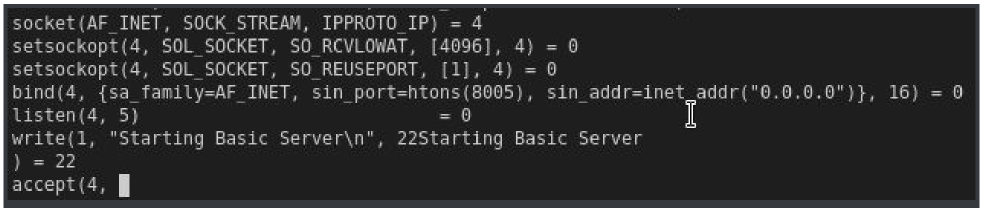
```
socket(AF_INET, SOCK_STREAM, IPPROTO_IP) = 6
fcntl(6, F_GETFL)                       = 0x2 (flags O_RDWR)
fcntl(6, F_SETFL, O_RDONLY|O_NONBLOCK)  = 0
setsockopt(6, SOL_SOCKET, SO_RCVLOWAT, [4096], 4) = 0
setsockopt(6, SOL_SOCKET, SO_REUSEPORT, [1], 4) = 0
bind(6, {sa_family=AF_INET, sin_port=htons(8005), sin_addr=inet_addr("0.0.0.0")}, 16) = 0
listen(6, 5)                            = 0
epoll_ctl(4, EPOLL_CTL_ADD, 5, {EPOLLIN, {u32=5, u64=4294967301}}) = 0
epoll_ctl(4, EPOLL_CTL_ADD, 6, {EPOLLIN, {u32=6, u64=4294967302}}) = 0
epoll_pwait(4,
```

<table>
<tr><td>Test#</td><td>4</td></tr>
<tr><td>Description</td><td>Ensure that select runs and uses non-blocking sockets.</td></tr>
<tr><td>Steps</td><td>● strace the server with select argument
○ Make sure server blocks on epoll_pwait and not on accept
○ Make sure socket is successfully set to SO_NONBLOCK and SO_REUSEPORT
○ Make sure SO_RCVLOWAT is set to BUFSIZE=4096</td></tr>
<tr><td>Result</td><td>
socket(AF_INET, SOCK_STREAM, IPPROTO_IP) = 5
fcntl(5, F_GETFL)   = 0x2 (flags O_RDWR)
fcntl(5, F_SETFL, O_RDONLY|O_NONBLOCK) = 0
setsockopt(5, SOL_SOCKET, SO_RCVLOWAT, [4096], 4) = 0
setsockopt(5, SOL_SOCKET, SO_REUSEPORT, [1], 4) = 0
bind(5, {sa_family=AF_INET, sin_port=htons(8005), sin_addr=inet_addr("0.0.0.0")}, 16) = 0
listen(5, 5)   = 0
select(64, [4 5], [], NULL, {tv_sec=59, tv_usec=743000}</td></tr>
<tr><td>Success</td><td>Yes</td></tr>
</table>

<table>
<tr><td>Test#</td><td>5</td></tr>
<tr><td>Description</td><td>Ensure that basic server waits on accept and has all correct flags set.</td></tr>
<tr><td>Steps</td><td>

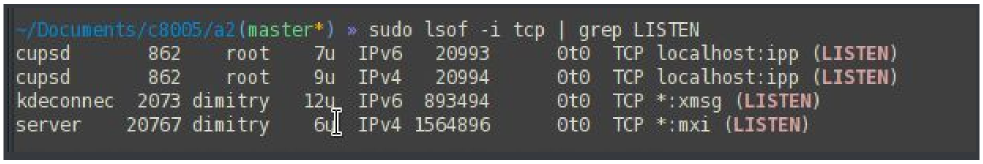
- strace the server with select argument
    - Make sure server blocks on accept
    - Make sure socket is successfully set to SO_REUSEPORT
    - Make sure SO_RCVLOWAT is set to BUFSIZE=4096

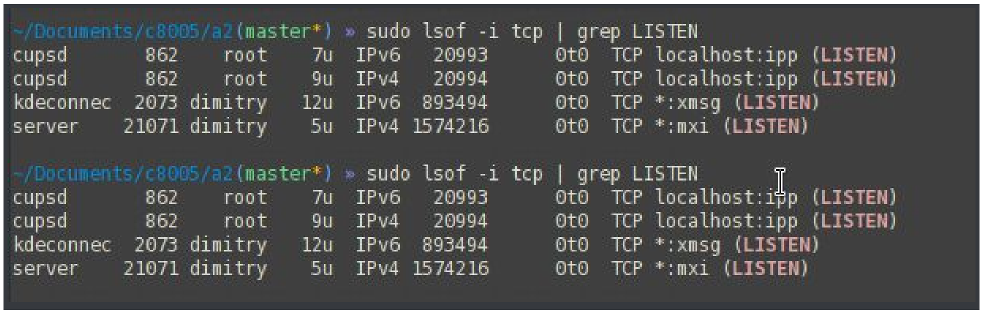
</td></tr>
<tr><td>Result</td><td>



```
socket(AF_INET, SOCK_STREAM, IPPROTO_IP) = 4
setsockopt(4, SOL_SOCKET, SO_RCVLOWAT, [4096], 4) = 0
setsockopt(4, SOL_SOCKET, SO_REUSEPORT, [1], 4) = 0
bind(4, {sa_family=AF_INET, sin_port=htons(8005), sin_addr=inet_addr("0.0.0.0")}, 16) = 0
listen(4, 5)                            = 0
write(1, "Starting Basic Server\n", 22Starting Basic Server
) = 22
accept(4, 
```

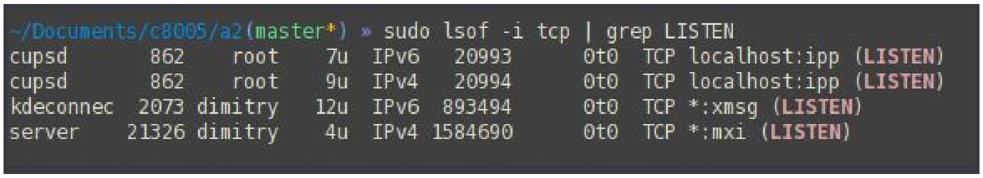
</td></tr>
<tr><td>Success</td><td>Yes</td></tr>
</table>

<table>
<tr><td>Test#</td><td>6</td></tr>
<tr><td>Description</td><td>Make sure that after 10 seconds of execution epoll server does not have any LISTEN sockets. All must be ESTABLISHED</td></tr>
<tr><td>Steps</td><td>● Run epoll server
● Run 512 clients for > 10 seconds
● After 10 seconds run sudo lsof -i tcp | grep LISTEN
● Check for server process</td></tr>
<tr><td>Result</td><td>
<pre>
~/Documents/c8005/a2(master*) » sudo lsof -i tcp | grep LISTEN
cupsd   862   root   7u IPv6   20993   0t0 TCP localhost:ipp (LISTEN)
cupsd   862   root   9u IPv4   20994   0t0 TCP localhost:ipp (LISTEN)
kdeconnec   2073 dimitry   12u IPv6 893494   0t0 TCP *:xmsg (LISTEN)
server   20767 dimitry   6u IPv4 1564896   0t0 TCP *:mxi (LISTEN)
</pre></td></tr>
<tr><td>Success</td><td>All but one socket function correctly. ~ Library Bug</td></tr>
</table>

Test#	7
Description	Make sure that after 10 seconds of execution select server does not have any LISTEN sockets. All must be ESTABLISHED
Steps	● Run select server ● Run 512 clients for > 10 seconds ● After 10 seconds run sudo lsof -i tcp | grep LISTEN ● Check for server process
Result	
Success	All but one socket function correctly. ~ Library Bug

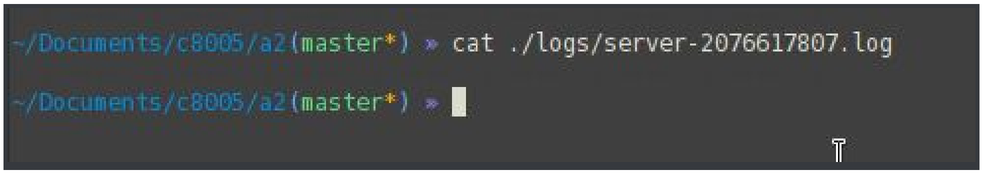
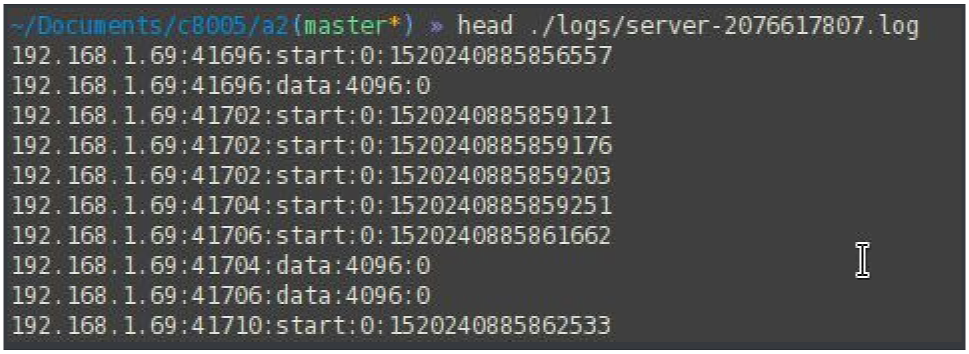
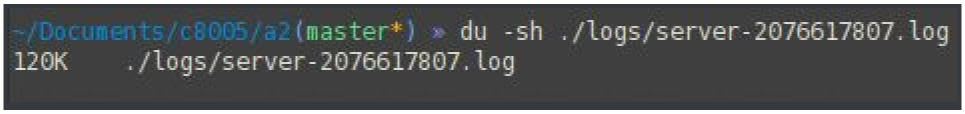
```
~/Documents/c8005/a2(master*) » sudo lsof -i tcp | grep LISTEN
cupsd       862     root    7u  IPv6   20993      0t0  TCP localhost:ipp (LISTEN)
cupsd       862     root    9u  IPv4   20994      0t0  TCP localhost:ipp (LISTEN)
kdeconnec  2073 dimitry    12u  IPv6  893494      0t0  TCP *:xmsg (LISTEN)
server    21071 dimitry     5u  IPv4 1574216      0t0  TCP *:mxi (LISTEN)

~/Documents/c8005/a2(master*) » sudo lsof -i tcp | grep LISTEN
cupsd       862     root    7u  IPv6   20993      0t0  TCP localhost:ipp (LISTEN)
cupsd       862     root    9u  IPv4   20994      0t0  TCP localhost:ipp (LISTEN)
kdeconnec  2073 dimitry    12u  IPv6  893494      0t0  TCP *:xmsg (LISTEN)
server    21071 dimitry     5u  IPv4 1574216      0t0  TCP *:mxi (LISTEN)
```

Test#	8
Description	Make sure that after 10 seconds of execution basic server does not have any LISTEN sockets. All must be ESTABLISHED
Steps	● Run basic server ● Run 512 clients for > 10 seconds ● After 10 seconds run sudo lsof -i tcp | grep LISTEN ● Check for server process
Result	
Success	ll but one socket function correctly. ~ Library Bug

```
~/Documents/c8005/a2(master*) » sudo lsof -i tcp | grep LISTEN
cupsd        862     root    7u  IPv6   20993      0t0  TCP localhost:ipp (LISTEN)
cupsd        862     root    9u  IPv4   20994      0t0  TCP localhost:ipp (LISTEN)
kdeconnec   2073 dimitry    12u  IPv6  893494      0t0  TCP *:xmsg (LISTEN)
server     21326 dimitry     4u  IPv4 1584690      0t0  TCP *:mxi (LISTEN)
```

Test#	9
Description	Ensure that the epoll server outputs to a log file when clients connect.
Steps	● Cat the log before starting connections and after. ● If the server populates the log then logging is working.

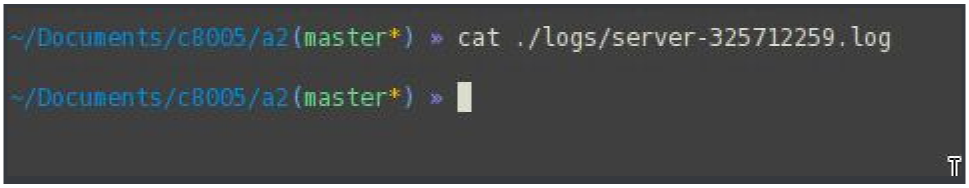
Result	
	`~/Documents/c8005/a2(master*) » cat ./logs/server-2076617807.log`
	`~/Documents/c8005/a2(master*) »`
	
	`~/Documents/c8005/a2(master*) » head ./logs/server-2076617807.log`
	`192.168.1.69:41696:start:0:1520240885856557`
	`192.168.1.69:41696:data:4096:0`
	`192.168.1.69:41702:start:0:1520240885859121`
	`192.168.1.69:41702:start:0:1520240885859176`
	`192.168.1.69:41702:start:0:1520240885859203`
	`192.168.1.69:41704:start:0:1520240885859251`
	`192.168.1.69:41706:start:0:1520240885861662`
	`192.168.1.69:41704:data:4096:0`
	`192.168.1.69:41706:data:4096:0`
	`192.168.1.69:41710:start:0:1520240885862533`
	
	`~/Documents/c8005/a2(master*) » du -sh ./logs/server-2076617807.log`
	`120K    ./logs/server-2076617807.log`

Success	Yes

<table>
<tr><td>Test#</td><td>10</td></tr>
<tr><td>Description</td><td>Ensure that the select server outputs to a log file when clients connect.</td></tr>
<tr><td>Steps</td><td>

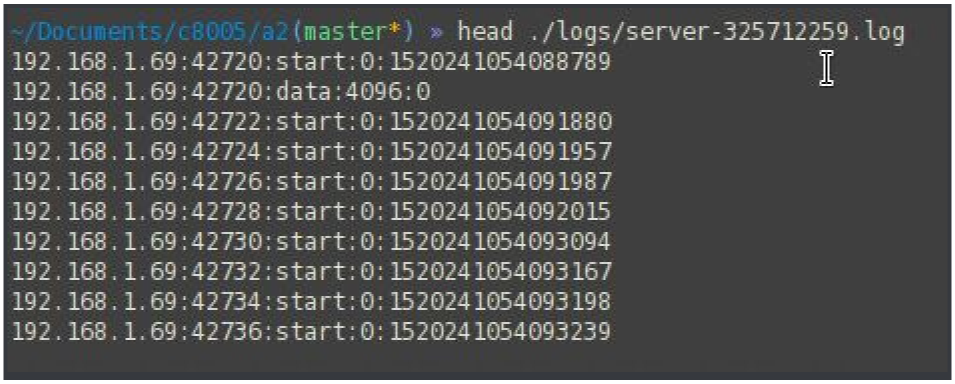
- Cat the log before starting connections and after.
- If the server populates the log then logging is working.

</td></tr>
<tr><td>Result</td><td>

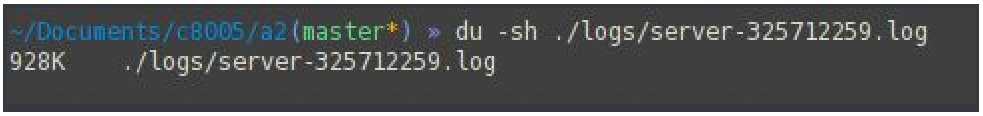


```
~/Documents/c8005/a2(master*) » cat ./logs/server-325712259.log

~/Documents/c8005/a2(master*) » 
```



```
~/Documents/c8005/a2(master*) » head ./logs/server-325712259.log
192.168.1.69:42720:start:0:1520241054088789
192.168.1.69:42720:data:4096:0
192.168.1.69:42722:start:0:1520241054091880
192.168.1.69:42724:start:0:1520241054091957
192.168.1.69:42726:start:0:1520241054091987
192.168.1.69:42728:start:0:1520241054092015
192.168.1.69:42730:start:0:1520241054093094
192.168.1.69:42732:start:0:1520241054093167
192.168.1.69:42734:start:0:1520241054093198
192.168.1.69:42736:start:0:1520241054093239
```



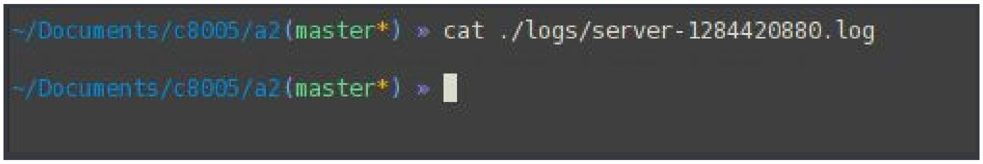
```
~/Documents/c8005/a2(master*) » du -sh ./logs/server-325712259.log
928K	./logs/server-325712259.log
```

</td></tr>
<tr><td>Success</td><td>Yes</td></tr>
</table>

<table>
<tr><td>Test#</td><td>11</td></tr>
<tr><td>Description</td><td>Ensure that the basic server outputs to a log file when clients connect.</td></tr>
<tr><td>Steps</td><td>

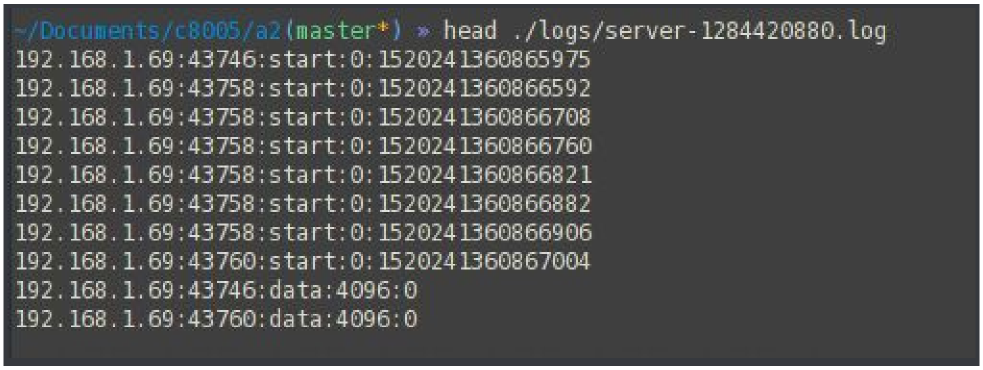
- Cat the log before starting connections and after.
- If the server populates the log then logging is working.

</td></tr>
<tr><td>Result</td><td>

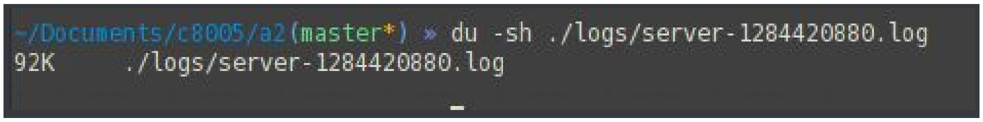


```
~/Documents/c8005/a2(master*) » cat ./logs/server-1284420880.log

~/Documents/c8005/a2(master*) » 
```



```
~/Documents/c8005/a2(master*) » head ./logs/server-1284420880.log
192.168.1.69:43746:start:0:1520241360865975
192.168.1.69:43758:start:0:1520241360866592
192.168.1.69:43758:start:0:1520241360866708
192.168.1.69:43758:start:0:1520241360866760
192.168.1.69:43758:start:0:1520241360866821
192.168.1.69:43758:start:0:1520241360866882
192.168.1.69:43758:start:0:1520241360866906
192.168.1.69:43760:start:0:1520241360867004
192.168.1.69:43746:data:4096:0
192.168.1.69:43760:data:4096:0
```



```
~/Documents/c8005/a2(master*) » du -sh ./logs/server-1284420880.log
92K     ./logs/server-1284420880.log
```

</td></tr>
<tr><td>Success</td><td>Yes</td></tr>
</table>