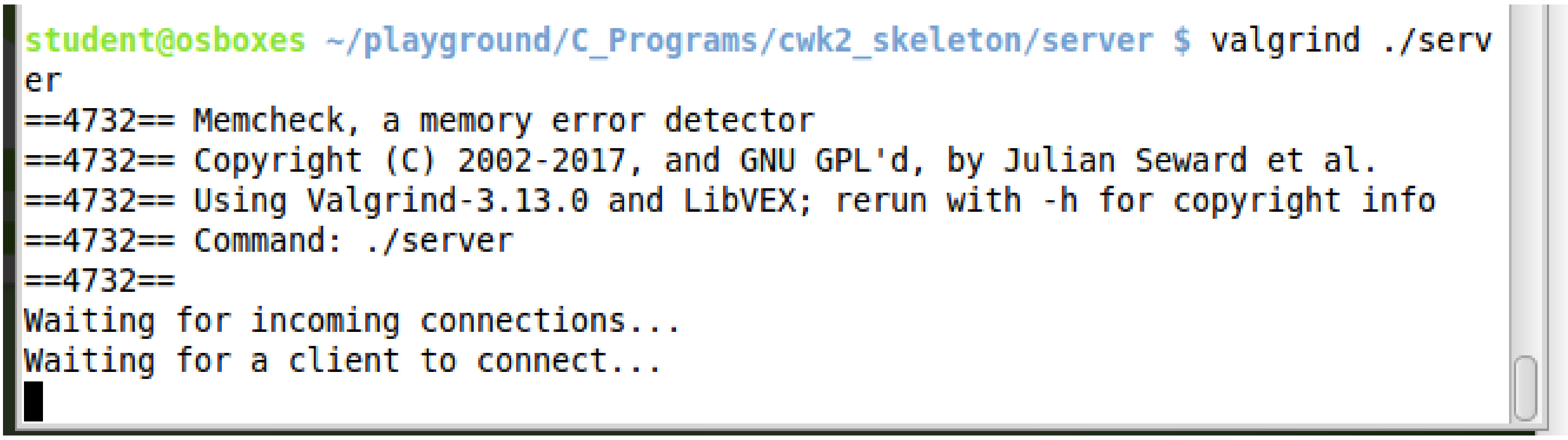
Coursework 2: Test Documentation

Test 1:

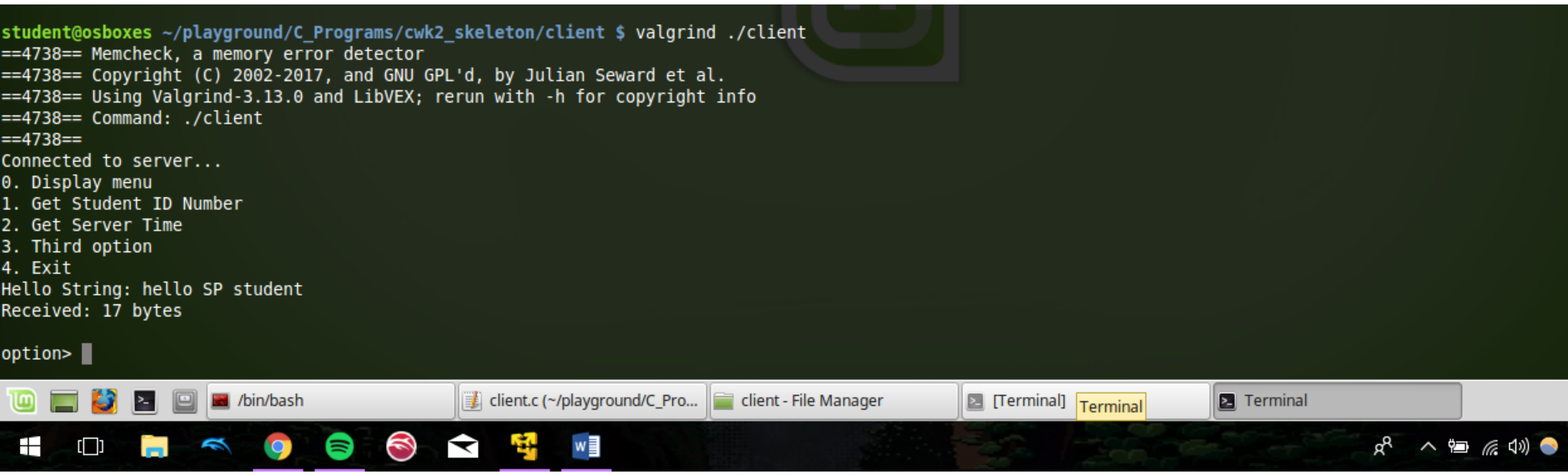
Test	Connecting Client to Server
Expected Result	The server will wait until the client connects. The server will generate a thread for the client and assign them a handler when they are connected. The client will display the menu when connected to the server.
Actual Result	Test ran as expected with no surprises.

Server



```
student@osboxes ~/playground/C_Programs/cwk2_skeleton/server $ valgrind ./serv
er
==4732== Memcheck, a memory error detector
==4732== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==4732== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==4732== Command: ./server
==4732==
Waiting for incoming connections...
Waiting for a client to connect...
```

Client

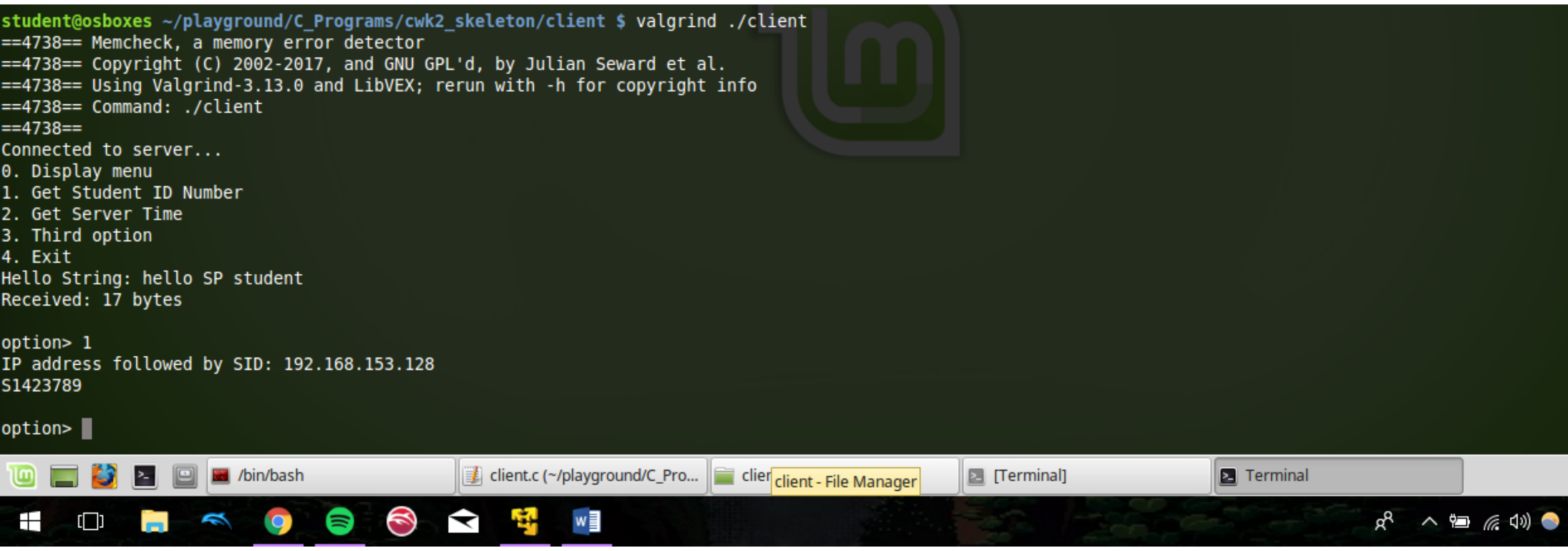


```
student@osboxes ~/playground/C_Programs/cwk2_skeleton/client $ valgrind ./client
==4738== Memcheck, a memory error detector
==4738== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==4738== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==4738== Command: ./client
==4738==
Connected to server...
0. Display menu
1. Get Student ID Number
2. Get Server Time
3. Third option
4. Exit
Hello String: hello SP student
Received: 17 bytes

option>
```

Test 2:

Test	Getting hardcoded student ID
Expected Result	The client will send the option ‘1’ to the server, and will then receive the ip address of the server and the student id number from the server.
Actual Result	Test ran as expected with no surprises

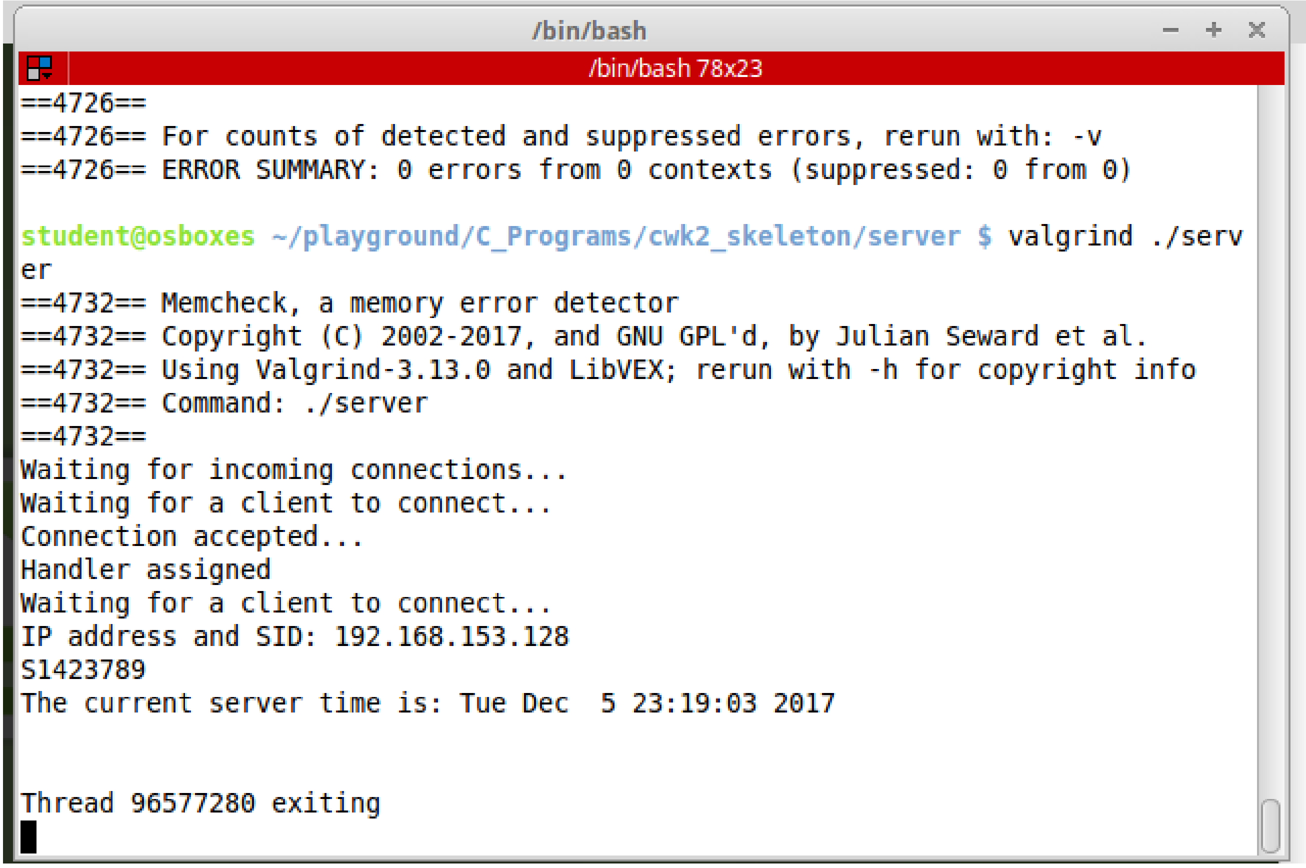
Client



```
student@osboxes ~/playground/C_Programs/cwk2_skeleton/client $ valgrind ./client
==4738== Memcheck, a memory error detector
==4738== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==4738== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==4738== Command: ./client
==4738==
Connected to server...
0. Display menu
1. Get Student ID Number
2. Get Server Time
3. Third option
4. Exit
Hello String: hello SP student
Received: 17 bytes

option> 1
IP address followed by SID: 192.168.153.128
S1423789

option> 
```

Server

NOTE: all functions currently call at the same time, this is because do-while loop has not been implemented yet!

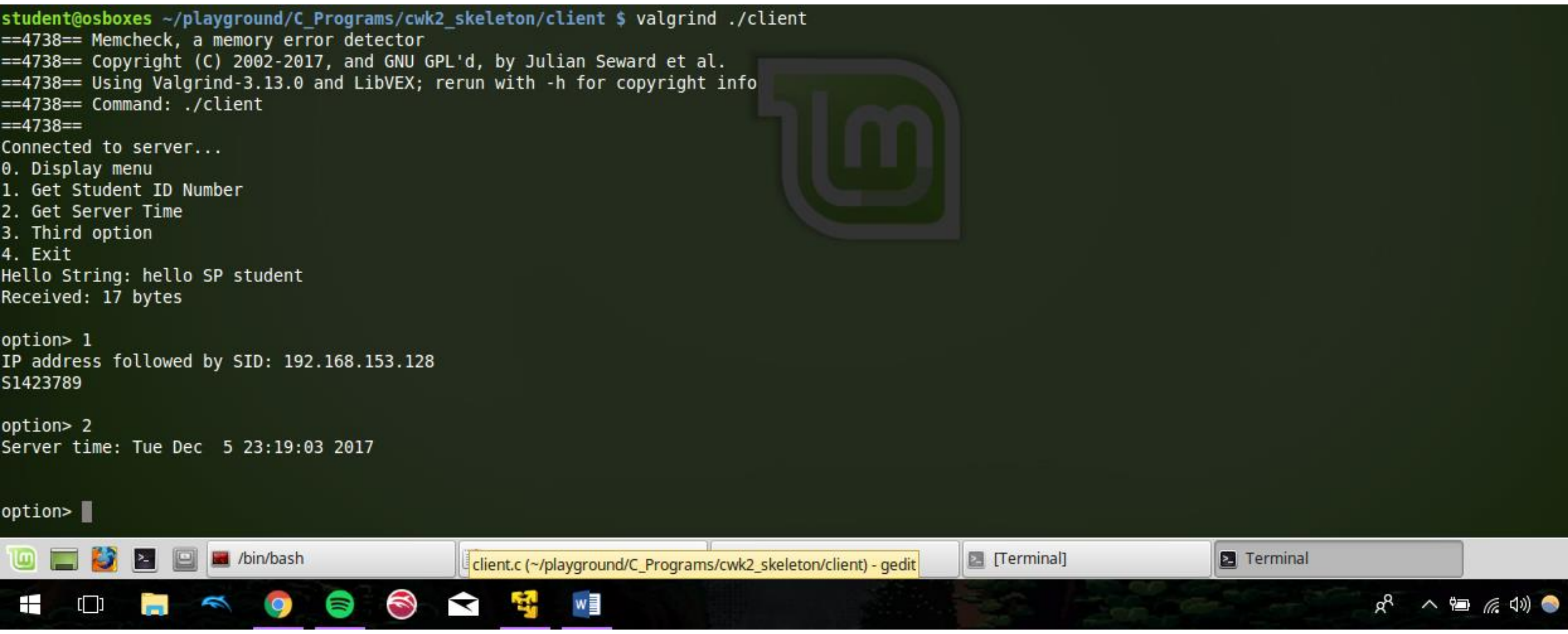


```
==4726==
==4726== For counts of detected and suppressed errors, rerun with: -v
==4726== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)

student@osboxes ~/playground/C_Programs/cwk2_skeleton/server $ valgrind ./serv
er
==4732== Memcheck, a memory error detector
==4732== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==4732== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==4732== Command: ./server
==4732==
Waiting for incoming connections...
Waiting for a client to connect...
Connection accepted...
Handler assigned
Waiting for a client to connect...
IP address and SID: 192.168.153.128
S1423789
The current server time is: Tue Dec  5 23:19:03 2017


Thread 96577280 exiting
```

Test 3:

Test	Getting current server time
Expected Result	The client will send the option ‘2’ to the server and will then receive the current server time from the server.
Actual Result	Test ran as expected with no surprises

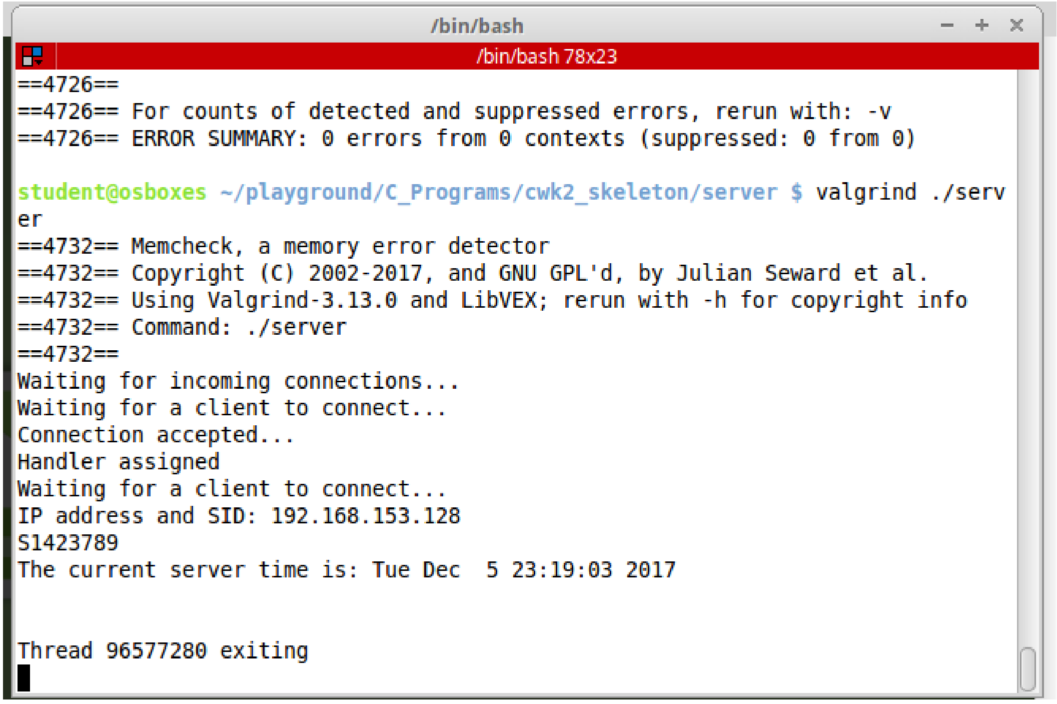
Client



```
student@osboxes ~/playground/C_Programs/cwk2_skeleton/client $ valgrind ./client
==4738== Memcheck, a memory error detector
==4738== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==4738== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==4738== Command: ./client
==4738==
Connected to server...
0. Display menu
1. Get Student ID Number
2. Get Server Time
3. Third option
4. Exit
Hello String: hello SP student
Received: 17 bytes

option> 1
IP address followed by SID: 192.168.153.128
S1423789

option> 2
Server time: Tue Dec  5 23:19:03 2017


option> 
```

/bin/bash
client.c (~/playground/C_Programs/cwk2_skeleton/client) - gedit
[Terminal]
Terminal

Server

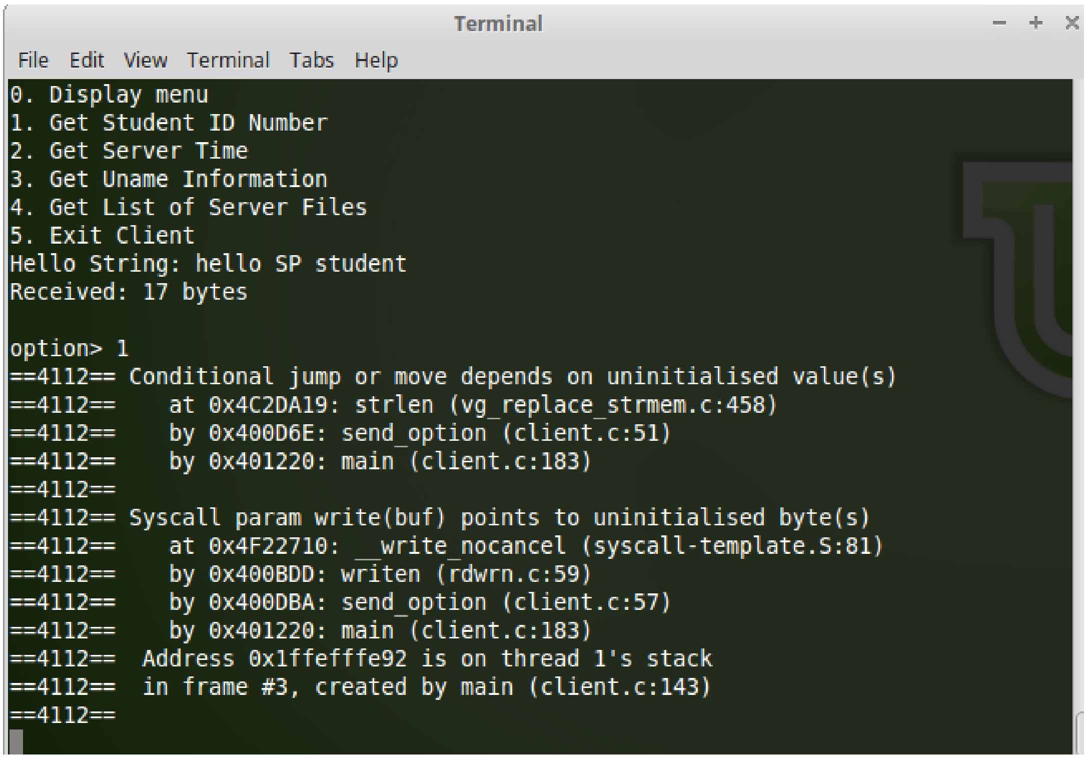


```
==4726==
==4726== For counts of detected and suppressed errors, rerun with: -v
==4726== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)

student@osboxes ~/playground/C_Programs/cwk2_skeleton/server $ valgrind ./serv
er
==4732== Memcheck, a memory error detector
==4732== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==4732== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==4732== Command: ./server
==4732==
Waiting for incoming connections...
Waiting for a client to connect...
Connection accepted...
Handler assigned
Waiting for a client to connect...
IP address and SID: 192.168.153.128
S1423789
The current server time is: Tue Dec  5 23:19:03 2017


Thread 96577280 exiting
```

Test 4.1:

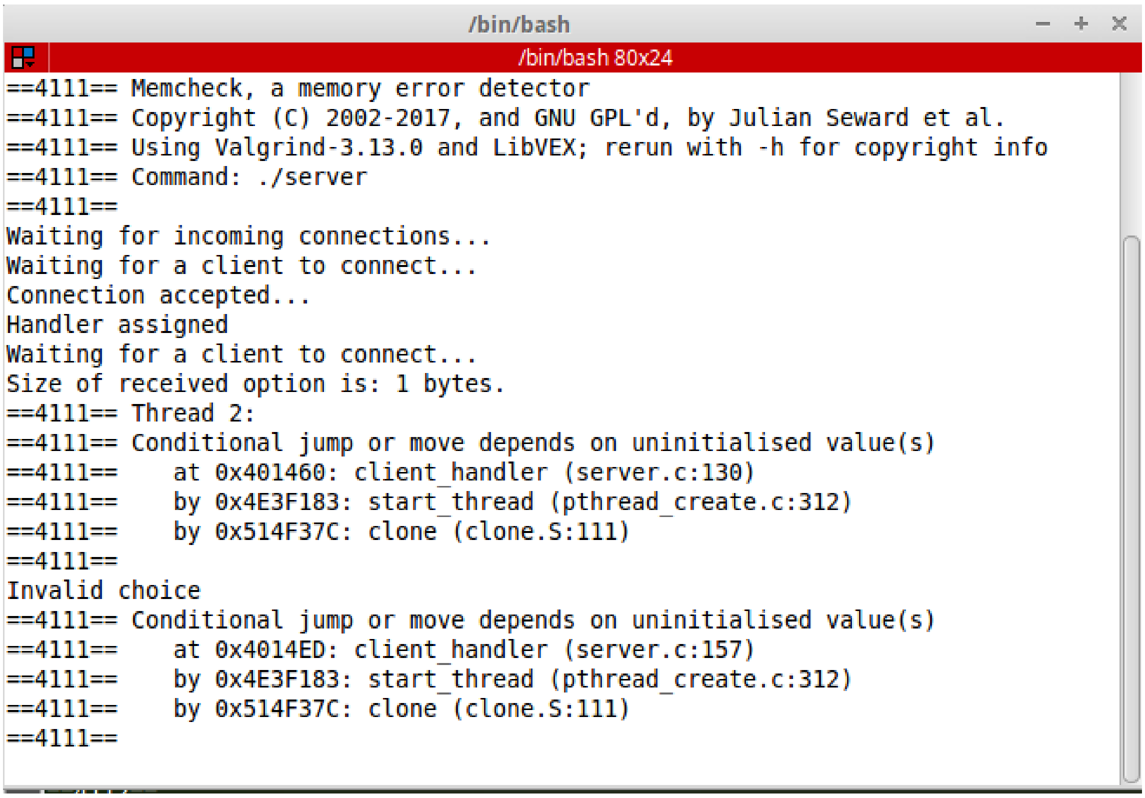
Test	Implementing do-while loop no.1
Expected Result	The client will be able to send the input option to the server, the appropriate function will then be called. For testing purposes, the client will enter option ‘1’.
Actual Result	Client crashed and valgrind gave back write errors, server displayed ‘invalid choice’. Valgrind also reported uninitialized values.

Client



```
0. Display menu
1. Get Student ID Number
2. Get Server Time
3. Get Uname Information
4. Get List of Server Files
5. Exit Client
Hello String: hello SP student
Received: 17 bytes

option> 1
==4112== Conditional jump or move depends on uninitialised value(s)
==4112==    at 0x4C2DA19: strlen (vg_replace_strmem.c:458)
==4112==    by 0x400D6E: send_option (client.c:51)
==4112==    by 0x401220: main (client.c:183)
==4112==
==4112== Syscall param write(buf) points to uninitialised byte(s)
==4112==    at 0x4F22710: __write_nocancel (syscall-template.S:81)
==4112==    by 0x400BDD: writen (rdwrn.c:59)
==4112==    by 0x400DBA: send_option (client.c:57)
==4112==    by 0x401220: main (client.c:183)
==4112==  Address 0x1ffefffe92 is on thread 1's stack
==4112==  in frame #3, created by main (client.c:143)
==4112==
```

Server

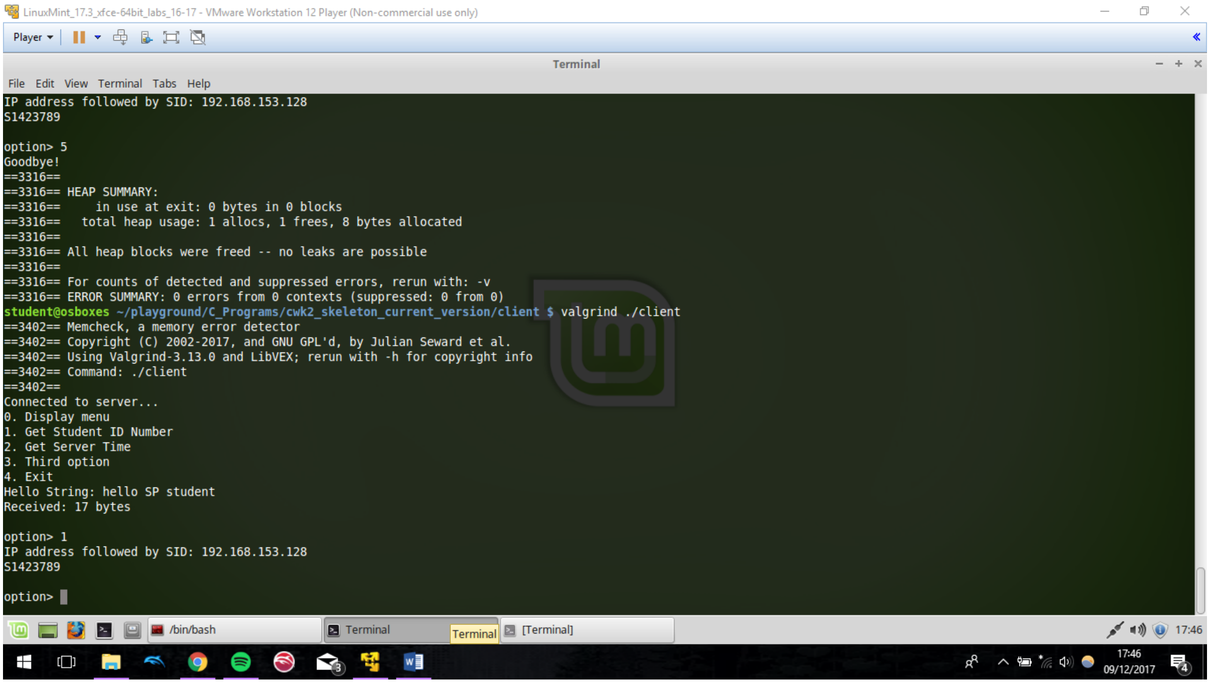


```
==4111== Memcheck, a memory error detector
==4111== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==4111== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==4111== Command: ./server
==4111==
Waiting for incoming connections...
Waiting for a client to connect...
Connection accepted...
Handler assigned
Waiting for a client to connect...
Size of received option is: 1 bytes.
==4111== Thread 2:
==4111== Conditional jump or move depends on uninitialised value(s)
==4111==    at 0x401460: client_handler (server.c:130)
==4111==    by 0x4E3F183: start_thread (pthread_create.c:312)
==4111==    by 0x514F37C: clone (clone.S:111)
==4111==
Invalid choice
==4111== Conditional jump or move depends on uninitialised value(s)
==4111==    at 0x4014ED: client_handler (server.c:157)
==4111==    by 0x4E3F183: start_thread (pthread_create.c:312)
==4111==    by 0x514F37C: clone (clone.S:111)
==4111==
```

Test 4.2:

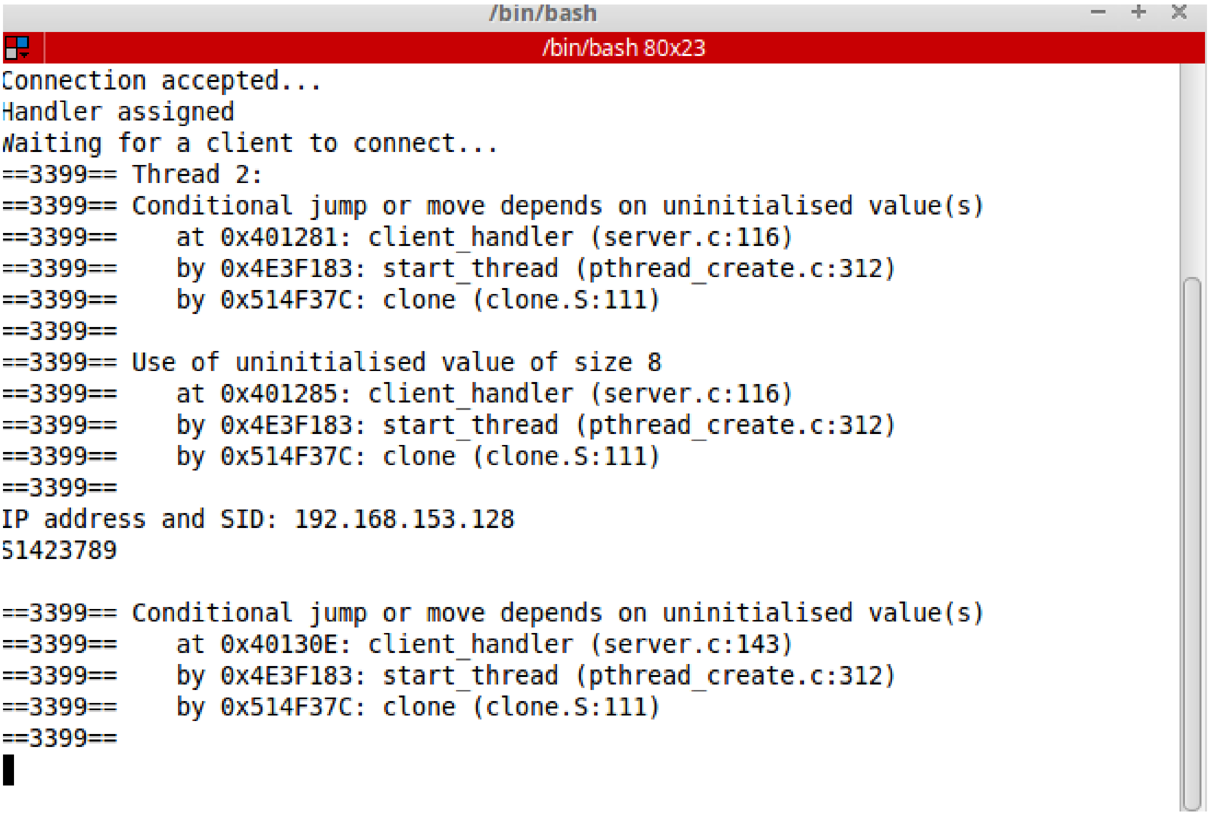
Test	Implementing do-while loop no.2
Expected Result	The client will be able to send the input option to the server, the appropriate function will then be called. For testing purposes, the client will enter option '1'.
Actual Result	Test ran as expected, but uninitialized values were still reported on the server side. The source of these errors could not be found.

Client


LinuxMint_17.3_xfce-64bit_labs_16-17 - VMware Workstation 12 Player (Non-commercial use only)
Player
Terminal
File Edit View Terminal Tabs Help
/bin/bash
Terminal
[Terminal]
17:46
09/12/2017

```
IP address followed by SID: 192.168.153.128
S1423789

option> 5
Goodbye!
==3316==
==3316== HEAP SUMMARY:
==3316==     in use at exit: 0 bytes in 0 blocks
==3316==   total heap usage: 1 allocs, 1 frees, 8 bytes allocated
==3316==
==3316== All heap blocks were freed -- no leaks are possible
==3316==
==3316== For counts of detected and suppressed errors, rerun with: -v
==3316== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
student@osboxes ~/playground/C_Programs/cwk2_skeleton_current_version/client $ valgrind ./client
==3402== Memcheck, a memory error detector
==3402== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==3402== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==3402== Command: ./client
==3402==
Connected to server...
0. Display menu
1. Get Student ID Number
2. Get Server Time
3. Third option
4. Exit
Hello String: hello SP student
Received: 17 bytes

option> 1
IP address followed by SID: 192.168.153.128
S1423789

option> 
```

Server

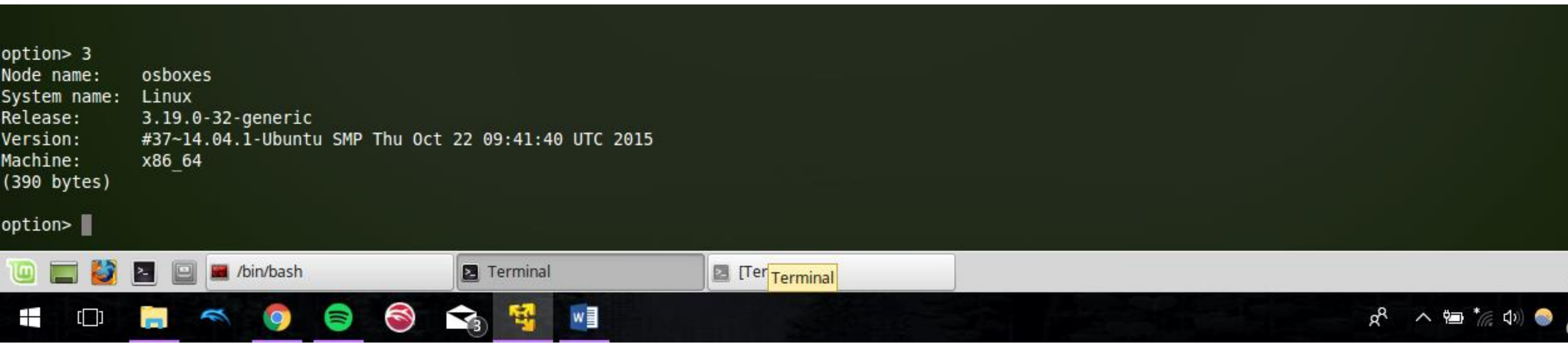


```
Connection accepted...
Handler assigned
Waiting for a client to connect...
==3399== Thread 2:
==3399== Conditional jump or move depends on uninitialised value(s)
==3399==    at 0x401281: client_handler (server.c:116)
==3399==    by 0x4E3F183: start_thread (pthread_create.c:312)
==3399==    by 0x514F37C: clone (clone.S:111)
==3399==
==3399== Use of uninitialised value of size 8
==3399==    at 0x401285: client_handler (server.c:116)
==3399==    by 0x4E3F183: start_thread (pthread_create.c:312)
==3399==    by 0x514F37C: clone (clone.S:111)
==3399==
IP address and SID: 192.168.153.128
51423789

==3399== Conditional jump or move depends on uninitialised value(s)
==3399==    at 0x40130E: client_handler (server.c:143)
==3399==    by 0x4E3F183: start_thread (pthread_create.c:312)
==3399==    by 0x514F37C: clone (clone.S:111)
==3399==
```

Test 5:

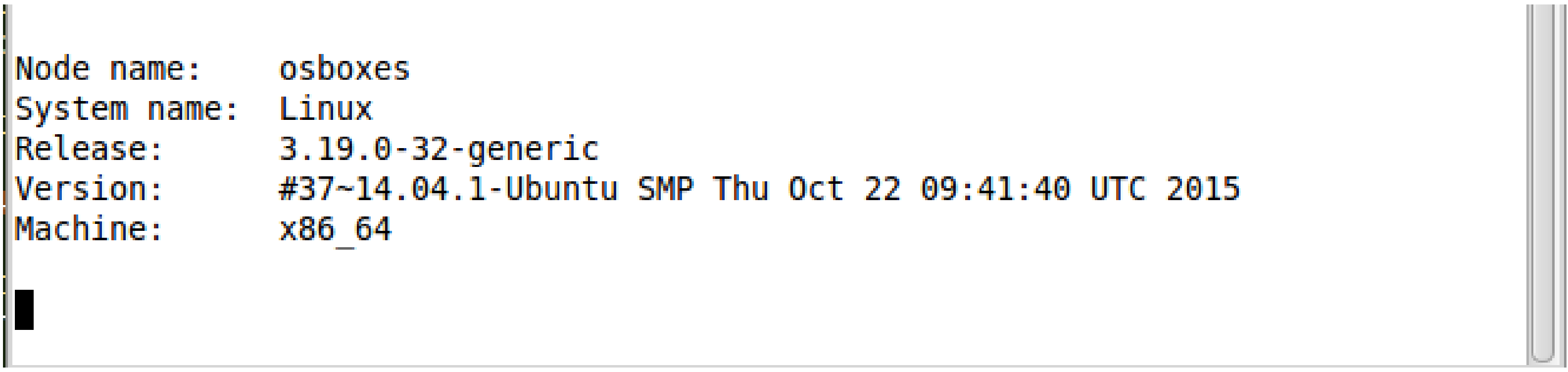
Test	Getting the server uname information
Expected Result	The client will input the option ‘3’ and receive the uname information from the server.
Actual Result	Test ran as expected with no surprises

Client



```
option> 3
Node name:     osboxes
System name:   Linux
Release:       3.19.0-32-generic
Version:       #37~14.04.1-Ubuntu SMP Thu Oct 22 09:41:40 UTC 2015
Machine:       x86_64
(390 bytes)

option> 
```

Server

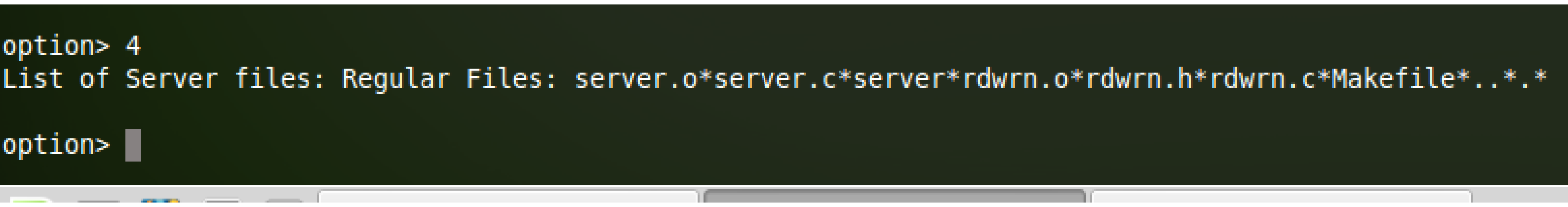


```
Node name:     osboxes
System name:   Linux
Release:       3.19.0-32-generic
Version:       #37~14.04.1-Ubuntu SMP Thu Oct 22 09:41:40 UTC 2015
Machine:       x86_64

```

Test 6:

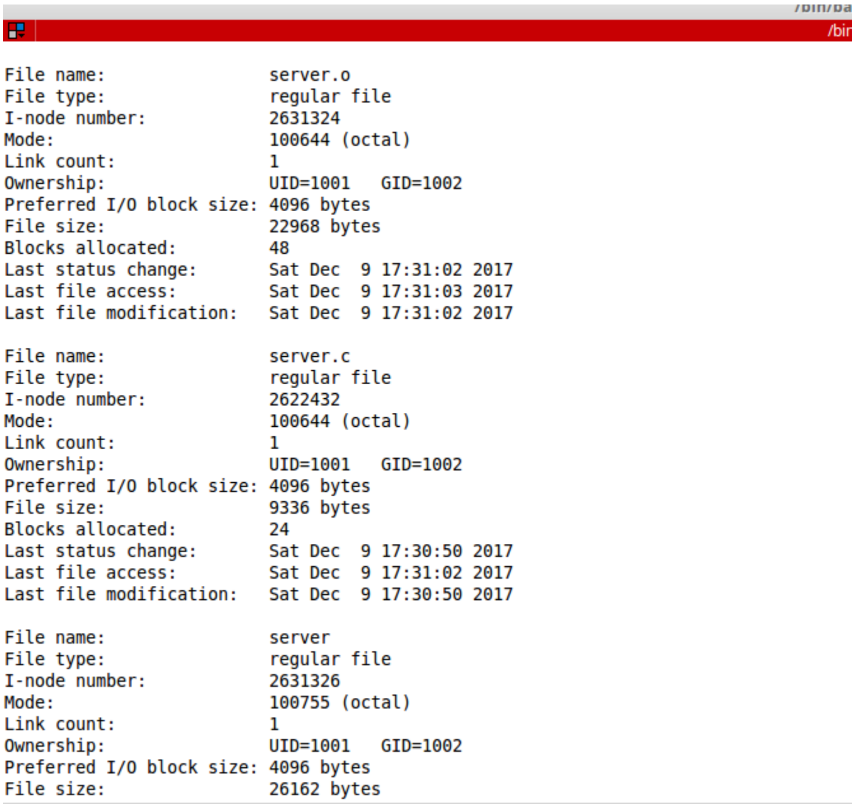
Test	Getting list of server files from the “upload” directory
Expected Result	The client will input the option ‘4’ and the server will generate a list of all files currently on its upload directory. This list will then be sent to the client as a string.
Actual Result	Test ran as expected with no surprises

Client

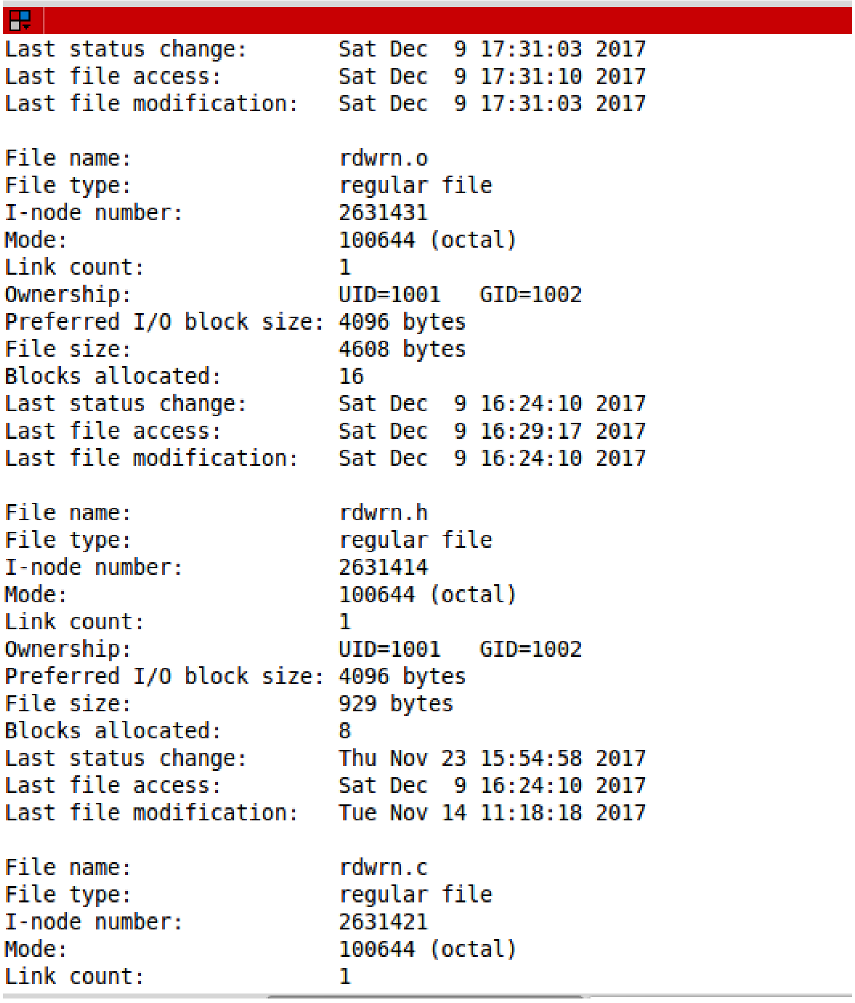


```
option> 4
List of Server files: Regular Files: server.o*server.c*server*rdwrn.o*rdwrn.h*rdwrn.c*Makefile*..*.*

option> 
```

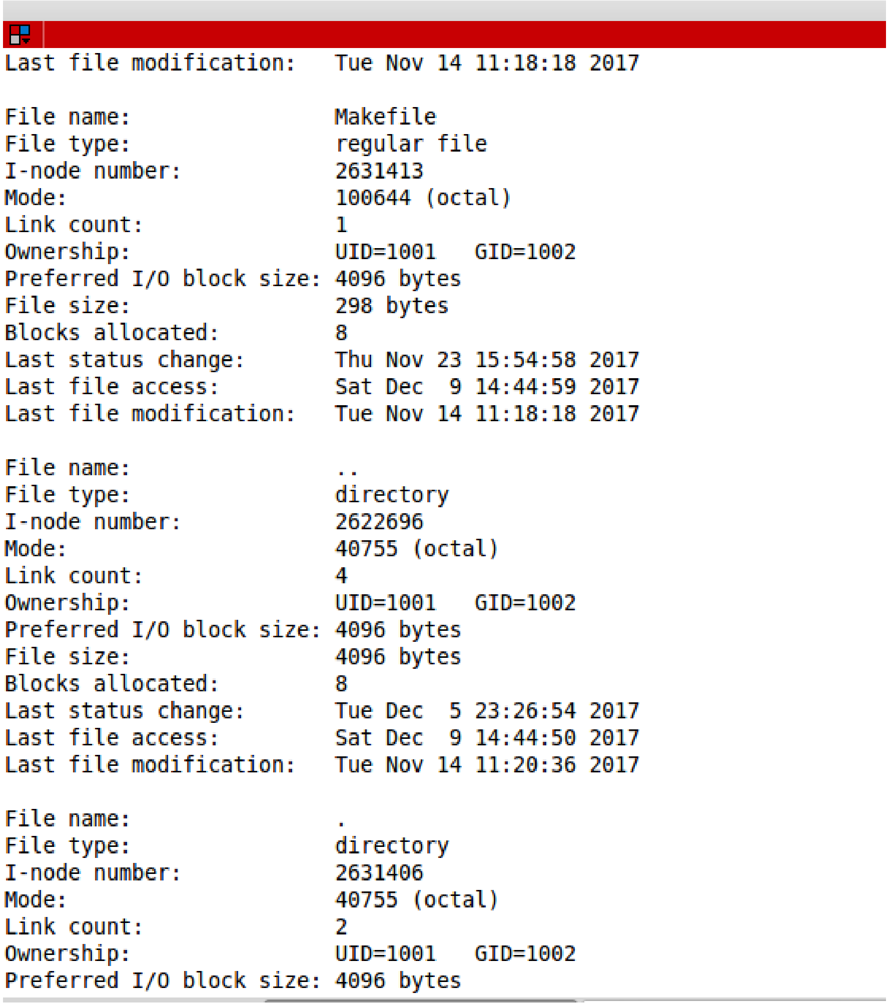
Server



```
File name:                server.o
File type:                regular file
I-node number:            2631324
Mode:                     100644 (octal)
Link count:               1
Ownership:                UID=1001   GID=1002
Preferred I/O block size: 4096 bytes
File size:                22968 bytes
Blocks allocated:         48
Last status change:       Sat Dec  9 17:31:02 2017
Last file access:         Sat Dec  9 17:31:03 2017
Last file modification:   Sat Dec  9 17:31:02 2017

File name:                server.c
File type:                regular file
I-node number:            2622432
Mode:                     100644 (octal)
Link count:               1
Ownership:                UID=1001   GID=1002
Preferred I/O block size: 4096 bytes
File size:                9336 bytes
Blocks allocated:         24
Last status change:       Sat Dec  9 17:30:50 2017
Last file access:         Sat Dec  9 17:31:02 2017
Last file modification:   Sat Dec  9 17:30:50 2017

File name:                server
File type:                regular file
I-node number:            2631326
Mode:                     100755 (octal)
Link count:               1
Ownership:                UID=1001   GID=1002
Preferred I/O block size: 4096 bytes
File size:                26162 bytes
```



```
Last status change:        Sat Dec  9 17:31:03 2017
Last file access:          Sat Dec  9 17:31:10 2017
Last file modification:    Sat Dec  9 17:31:03 2017

File name:                 rdwrn.o
File type:                 regular file
I-node number:             2631431
Mode:                      100644 (octal)
Link count:                1
Ownership:                 UID=1001   GID=1002
Preferred I/O block size: 4096 bytes
File size:                 4608 bytes
Blocks allocated:          16
Last status change:        Sat Dec  9 16:24:10 2017
Last file access:          Sat Dec  9 16:29:17 2017
Last file modification:    Sat Dec  9 16:24:10 2017

File name:                 rdwrn.h
File type:                 regular file
I-node number:             2631414
Mode:                      100644 (octal)
Link count:                1
Ownership:                 UID=1001   GID=1002
Preferred I/O block size: 4096 bytes
File size:                 929 bytes
Blocks allocated:          8
Last status change:        Thu Nov 23 15:54:58 2017
Last file access:          Sat Dec  9 16:24:10 2017
Last file modification:    Tue Nov 14 11:18:18 2017

File name:                 rdwrn.c
File type:                 regular file
I-node number:             2631421
Mode:                      100644 (octal)
Link count:                1
```



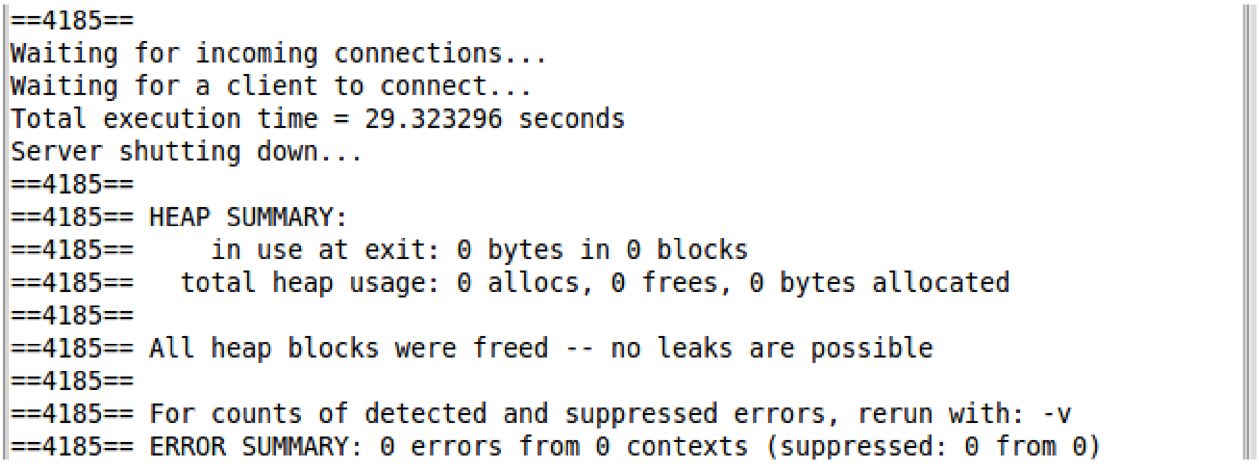
```
Last file modification:   Tue Nov 14 11:18:18 2017

File name:                Makefile
File type:                regular file
I-node number:            2631413
Mode:                     100644 (octal)
Link count:               1
Ownership:                UID=1001   GID=1002
Preferred I/O block size: 4096 bytes
File size:                298 bytes
Blocks allocated:         8
Last status change:       Thu Nov 23 15:54:58 2017
Last file access:         Sat Dec  9 14:44:59 2017
Last file modification:   Tue Nov 14 11:18:18 2017

File name:                ..
File type:                directory
I-node number:            2622696
Mode:                     40755 (octal)
Link count:               4
Ownership:                UID=1001   GID=1002
Preferred I/O block size: 4096 bytes
File size:                4096 bytes
Blocks allocated:         8
Last status change:       Tue Dec  5 23:26:54 2017
Last file access:         Sat Dec  9 14:44:50 2017
Last file modification:   Tue Nov 14 11:20:36 2017

File name:                .
File type:                directory
I-node number:            2631406
Mode:                     40755 (octal)
Link count:               2
Ownership:                UID=1001   GID=1002
Preferred I/O block size: 4096 bytes
```

Test 7:

Test	Getting total server execution time by sending a SIGTERM
Expected Result	The server will receive a SIGTERM and execute the code in its signal handler which gets the total execution time, and exits the server.
Actual Result	Test ran as expected with no surprises

Server

The following executes when “kill 4185” – the pid of the server – is entered



```
==4185==
Waiting for incoming connections...
Waiting for a client to connect...
Total execution time = 29.323296 seconds
Server shutting down...
==4185==
==4185== HEAP SUMMARY:
==4185==     in use at exit: 0 bytes in 0 blocks
==4185==   total heap usage: 0 allocs, 0 frees, 0 bytes allocated
==4185==
==4185== All heap blocks were freed -- no leaks are possible
==4185==
==4185== For counts of detected and suppressed errors, rerun with: -v
==4185== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```