



COMP4981 Assignment2 Test Cases

SAM LEE
A01029480

Table of Contents

Main Scenario.....	2
Scenario #1: Create a message queue and semaphore	2
Scenario #2: Read a message.....	2
Scenario #3: Fork when a message is received.....	3
Scenario #4: Open & read file.....	4
Scenario #5: Send & receive file content	5

Main Scenario

Scenario #1: Create a message queue and semaphore

Test steps

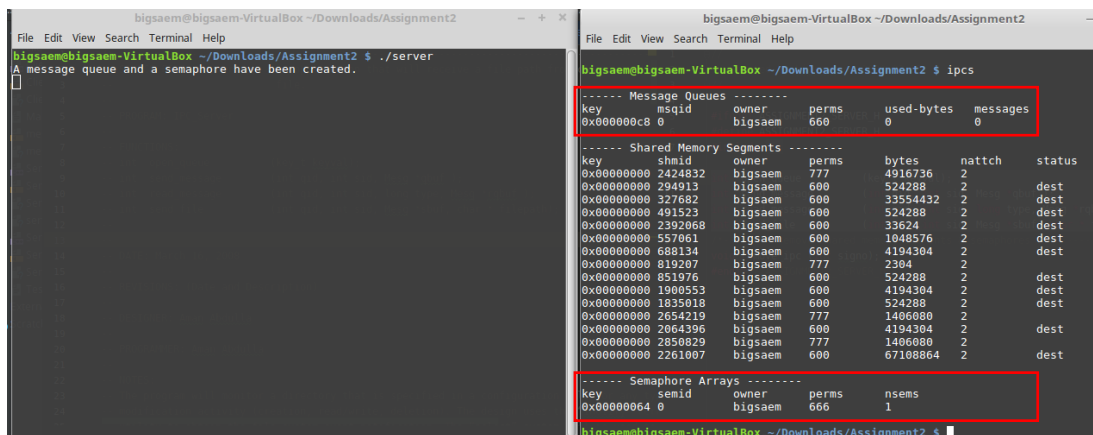
1. Start the application (server)
2. Type 'ipcs' in another terminal

Expected result

- A message queue and semaphore are created

Actual result

- A message queue and semaphore are created



The image shows two terminal windows from a VirtualBox environment. The left window shows the execution of a server program, which outputs a confirmation message. The right window shows the output of the 'ipcs' command, which lists system-wide IPC resources. Two sections of the output are highlighted with red boxes: 'Message Queues' and 'Semaphore Arrays'.

```
bigsaem@bigsaem-VirtualBox ~/Downloads/Assignment2
File Edit View Search Terminal Help
bigsaem@bigsaem-VirtualBox ~/Downloads/Assignment2 $ ./server
A message queue and a semaphore have been created.

bigsaem@bigsaem-VirtualBox ~/Downloads/Assignment2 $ ipcs
----- Message Queues -----
key      msqid    owner    perms    used-bytes  messages
0x000000c0 0        bigsaem  660      0            0

----- Shared Memory Segments -----
key      shmid    owner    perms    bytes       nattch     status
0x00000000 2424832  bigsaem  777      4916736     2          dest
0x00000000 294913   bigsaem  600      524288      2          dest
0x00000000 327682   bigsaem  600      33554432    2          dest
0x00000000 491523   bigsaem  600      524288      2          dest
0x00000000 2392068  bigsaem  600      33624       2          dest
0x00000000 557061   bigsaem  600      1048576     2          dest
0x00000000 688134   bigsaem  600      4194304     2          dest
0x00000000 819207   bigsaem  777      2304        2          dest
0x00000000 851976   bigsaem  600      524288      2          dest
0x00000000 1900593  bigsaem  600      4194304     2          dest
0x00000000 1835018  bigsaem  600      524288      2          dest
0x00000000 2654219  bigsaem  777      1406080     2          dest
0x00000000 2064396  bigsaem  600      4194304     2          dest
0x00000000 2850829  bigsaem  777      1406080     2          dest
0x00000000 2261007  bigsaem  600      67108864    2          dest

----- Semaphore Arrays -----
key      semid     owner    perms    nsems
0x00000064 0        bigsaem  666      1

bigsaem@bigsaem-VirtualBox ~/Downloads/Assignment2 $
```

Figure 1 message queue and semaphore are created

Result : PASS

Scenario #2: Read a message

Test steps

1. Set a break point on read function on server
2. Run a server
3. Run a client //it sends a message when it's executed
4. Check ipcs to see if a message is on message queue
5. Release the server from break point
6. Check ipcs to see if a message is read

Expected result

- A message is written on the queue and consumed by server

Actual result

- A message is written on the queue and consumed by server

- A child process is created

Actual result

- A child process is created

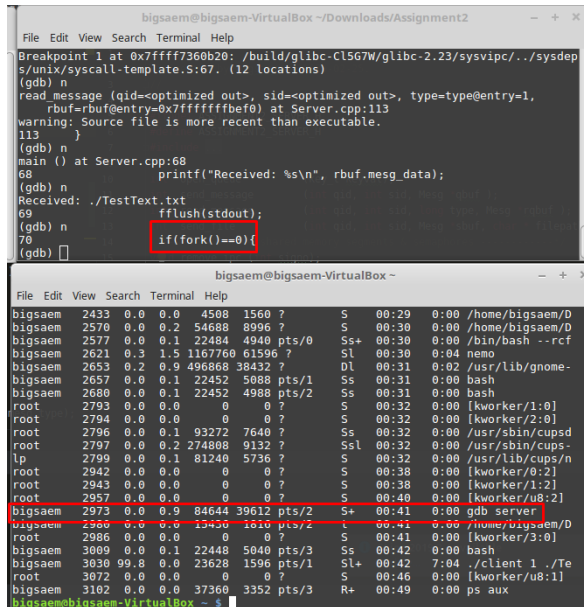


Figure 4 Fork

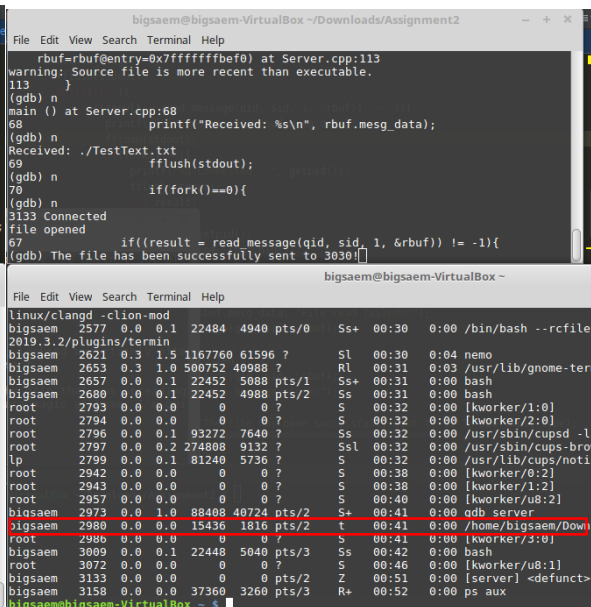


Figure 5 Child process is created

Result: PASS

Scenario #4: Open & read file

Test steps

1. Set a break point on file open function on server (ifstream)
2. Run a server
3. Run a client //it sends a message when it's executed
4. Release the server from break point
6. Check ifstream value
7. Check the length of the file by using ifstream.end
8. Compare actual length of the file outside of the program

Expected result

- File opened and the length is the same

Actual result

- File opened and the length is the same

```

bigsaem@bigsaem-VirtualBox ~/Downloads/Assignment2 $ ./server
A message queue and a semaphore have been created.
Received: ./TestText.txt
3413 Connected
file opened
448810The file has been successfully sent to 3411!

```

Figure 4 The recognized length of the file is the same as the actual length

Result:PASS

Scenario #5: Send & receive file content

Test steps

1. Run a server
2. Run a client
3. Compare the text that the client received to the original file

Expected result

- The text is the same as the original file

Actual result

- The text is the same as the original file

```

at the nerve of Harry, carrying an owl in a cage in a station full of ordinary p
eople. Behind him stood Aunt Petunia and Dudley, looking terrified at the very s
ight of Harry.

"You must be Harry's family!" said Mrs. Weasley.

"In a manner of speaking," said Uncle Vernon. "Hurry up, boy, we haven't got all
day." He walked away.

Harry hung back for a last word with Ron and Hermione.

"See you over the summer, then."

"Hope you have -- er -- a good holiday," said Hermione, looking uncertainly afte
r Uncle Vernon, shocked that anyone could be so unpleasant.

"Oh, I will," said Harry, and they were surprised at the grin that was spreading
over his face. "They don't know we're not allowed to use magic at home. I'm goi
ng to have a lot of fun with Dudley this summer...."

THE END

```

```

at the nerve of Harry, carrying an owl in a cage in a station full of ordinary
people. Behind him stood Aunt Petunia and Dudley, looking terrified at the very
sight of Harry.

"You must be Harry's family!" said Mrs. Weasley.

"In a manner of speaking," said Uncle Vernon. "Hurry up, boy, we haven't got all
day." He walked away.

Harry hung back for a last word with Ron and Hermione.

"See you over the summer, then."

"Hope you have -- er -- a good holiday," said Hermione, looking uncertainly after
Uncle Vernon, shocked that anyone could be so unpleasant.

"Oh, I will," said Harry, and they were surprised at the grin that was spreading
over his face. "They don't know we're not allowed to use magic at home. I'm going
to have a lot of fun with Dudley this summer...."

THE END |

```

Result:PASS