COMP4981 Assignment2 Test Cases

SAM LEE
A01029480

Table of Contents

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

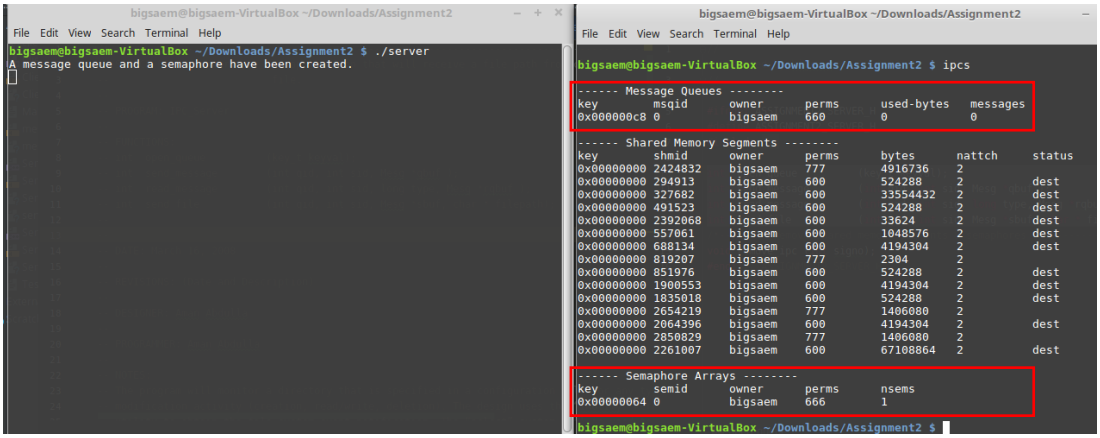


Figure 1 message queue and semaphore are created

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

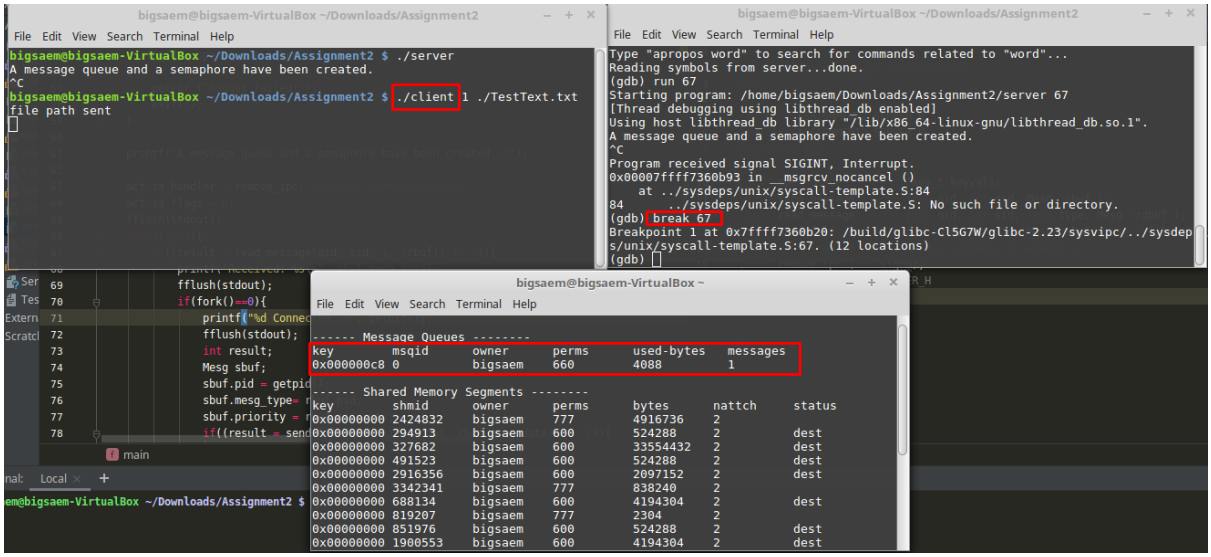


Figure 2 Client sent a message

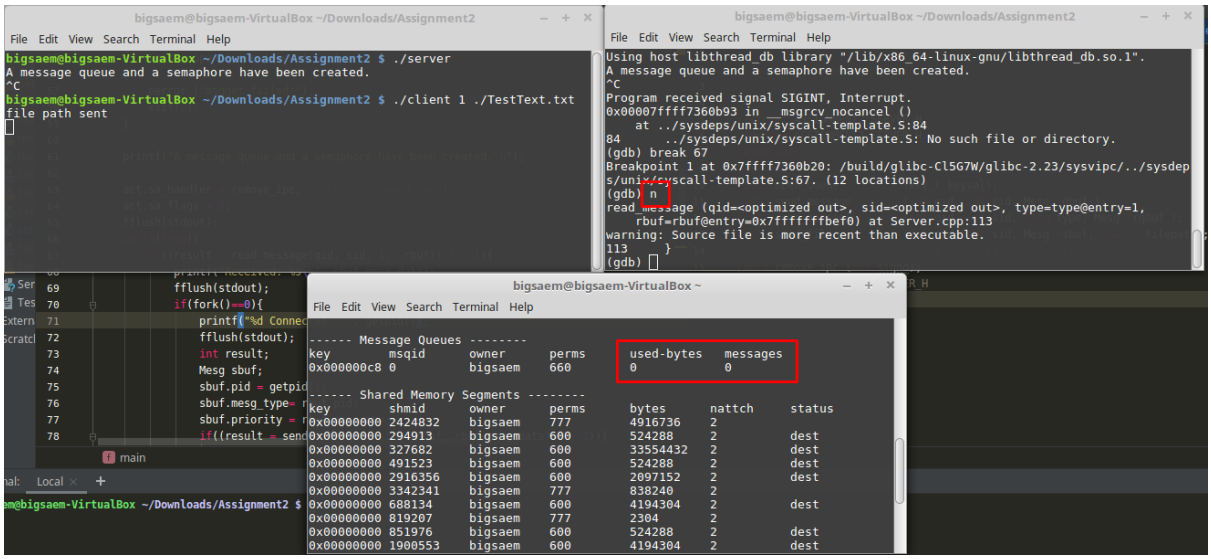


Figure 3 Server read a message

Result: PASS

Scenario #3: Fork when a message is received

Test steps

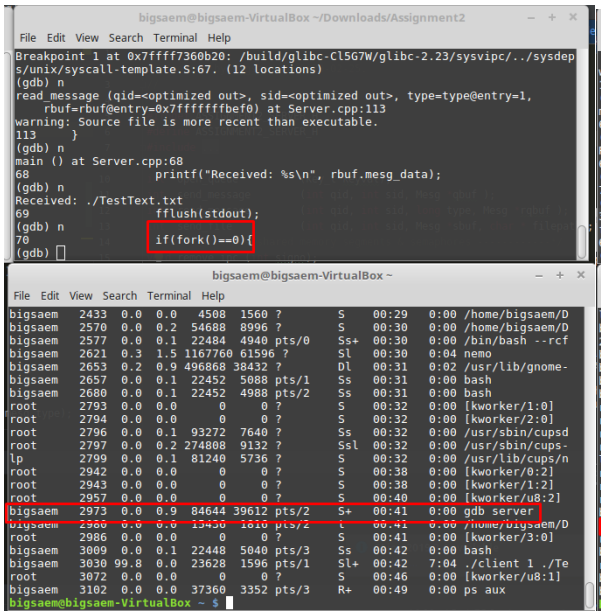
1. Set a break point on fork function on server
2. Run a server
3. Run a client //it sends a message when it's executed
4. Check ps aux to see what processes are running
5. Release the server from break point
6. Check ps aux to see if the child process is created

Expected result

- A child process is created

Actual result

- A child process is created



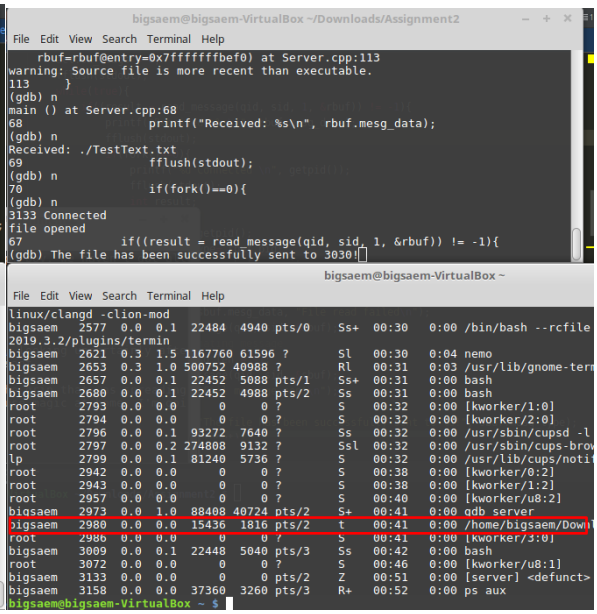


Figure 4 Fork

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

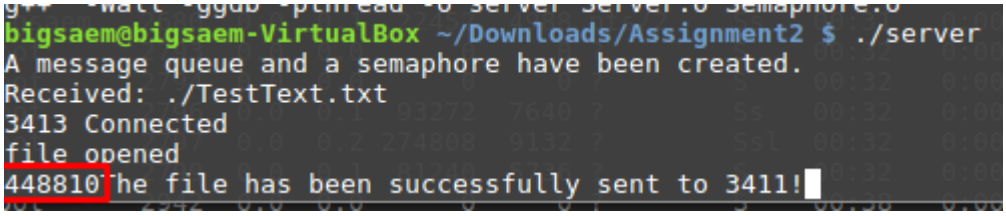


Figure 4 The recognized length of the file is the same as the actual length

Result:PASS

Scenario #5: Send & receive file content

Test steps

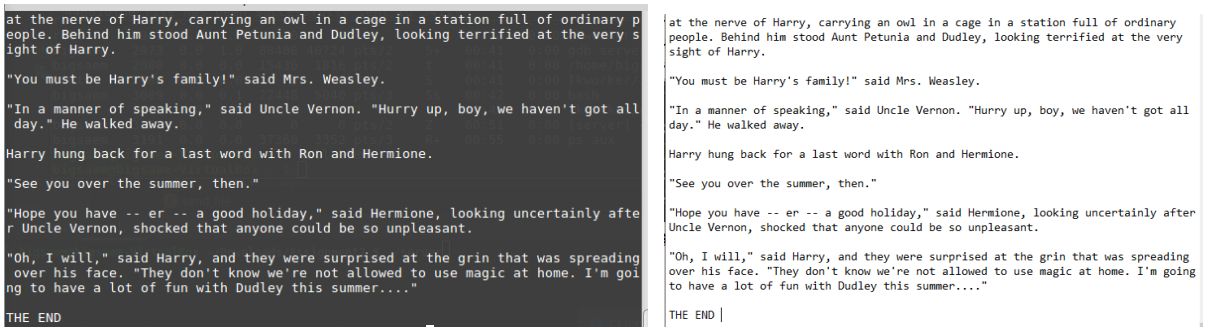
1. Run a server
2. Run a client
3. Compare the text that the client received to the original file

Expected result

- The text is the same as the original file

Actual result

- The text is the same as the original file



Result:PASS