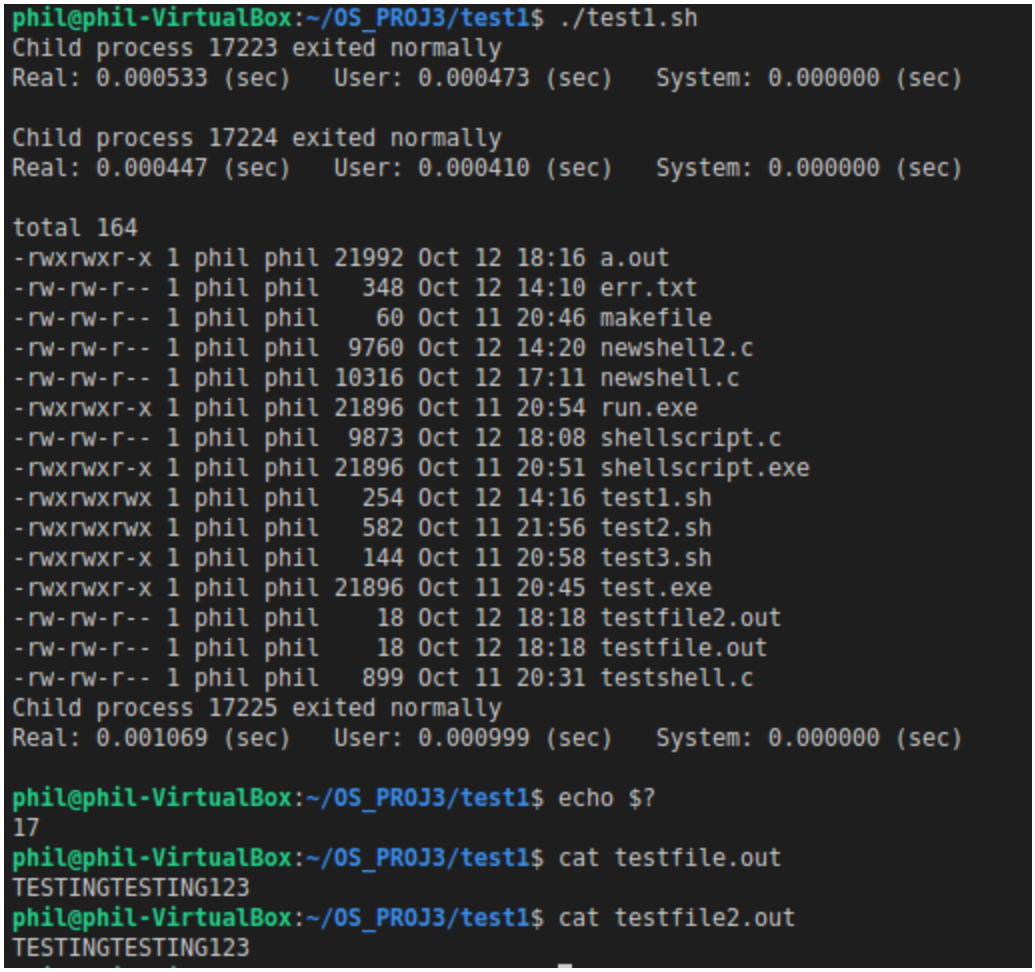
OS Problem set3 Q3 Documentation

Problem 3 - Simple Shell Program

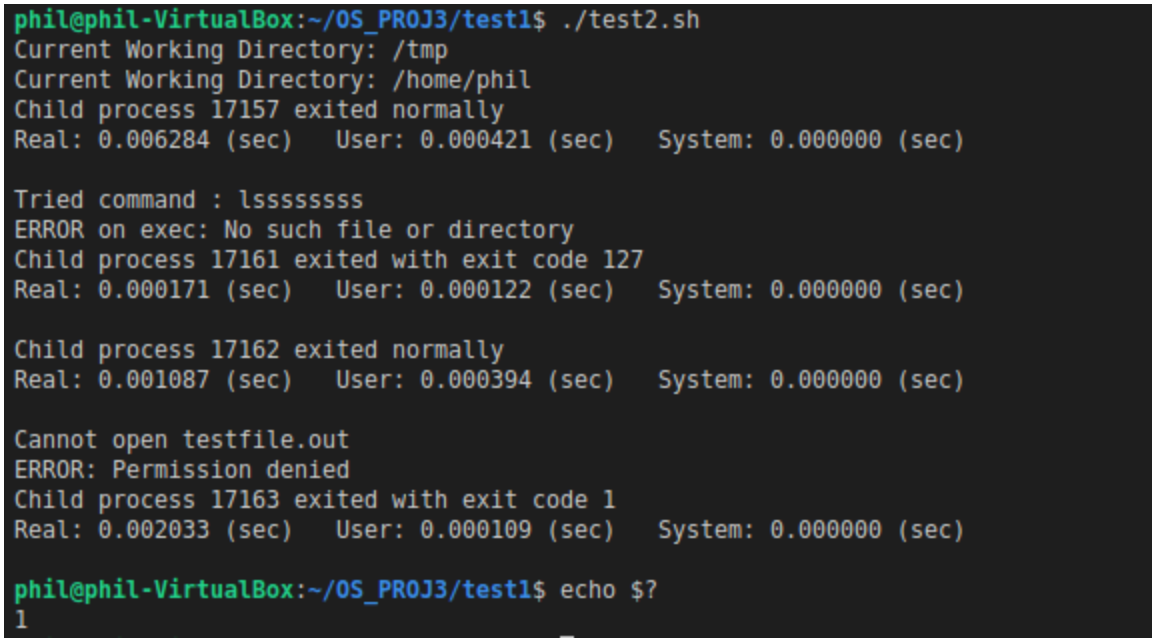
Thodori Kapouranis, Brian Doan, Philip Blumin

Test Case 1 (test1.sh):



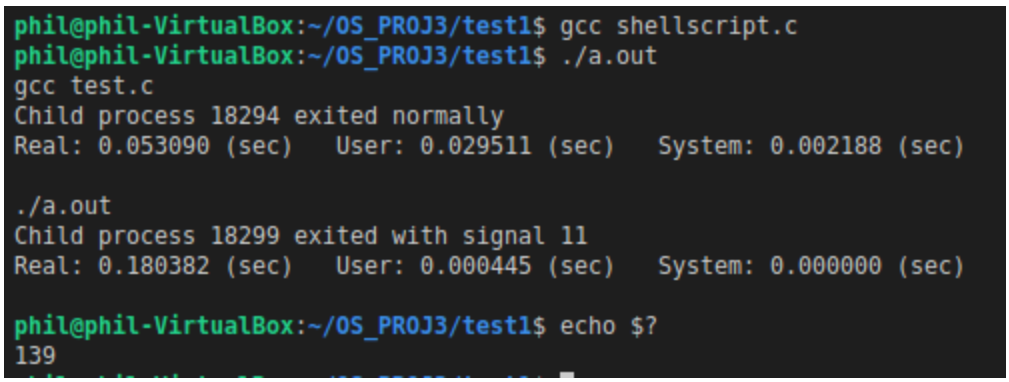
The image above is a run of test1.sh. After each line in the script, the program returns the times for the process. The real time is greater than the sum of the user and system times. The exit status is 17 after testing with echo $?

Test Case 2 (test2.sh):



The above image is the test run for test2.sh. The pwd command implementation works as the program shows the correct working directory after the respective cd line of the shell script. Similarly, the cd command works since the directory change is displayed. The lssssssss will not work but will not stop the program as shown. The exit status is 1 since the redirection is supposed to fail.

Test Case 3 (Signal Error):



The above image is a test run of a program that returns a segmentation fault inside the shell script. As seen the child process exits with signal 11. Doing the command echo$? after returns 139 as expected because this is the sum of the child exit status 11 and the 128 exit status.

Test Case 4 (Using the shell to run system calls):

```
./file_checker.exe / 2>error.txt

  ---------FILE TYPE COUNT---------
  Named pipe (FIFO):            18
  Character Device:             0
  Directory:                    25827
  Block Device:                 0
  Regular File:                 156234
  Symlink:                      29850
  Network/IPC socket:           7
  Bad Name count:               3

  ----------REGULAR FILES----------
  Size of files (in bytes):     13480381586
  Number of blocks:             3377882

  ----------LINK STATS-------------
  Files with multiple links:    80
  Invalid symlinks:             13363

  Child process 17627 exited normally
Real: 0.748226 (sec)   User: 0.140840 (sec)   System: 0.508588 (sec)
```

Above is an example of a command that should have a higher CPU System time than the user. The program used a lot of system calls so it is expected that the child will spend a significant amount of time in kernel mode rather than user mode.

Test Case 5 (Redirecting into a file that doesn't exist):

```
phil@phil-VirtualBox:~/OS_PROJ3/test1$ ./a.out
echo <DoesNotExist
Cannot open DoesNotExist
ERROR: No such file or directory
Child process 17861 exited with exit code 1
Real: 0.000194 (sec)   User: 0.000121 (sec)   System: 0.000000 (sec)

phil@phil-VirtualBox:~/OS_PROJ3/test1$ echo $?
1
```

Above is an example of an attempt to redirect a file that doesn't not exist. As seen the appropriate error message and exit code is returned. The shell does not exit on the error, a ctrl+D EOF character is sent.