Zhang Jiarong
120090495

CSC3150 Assignment 1 Report

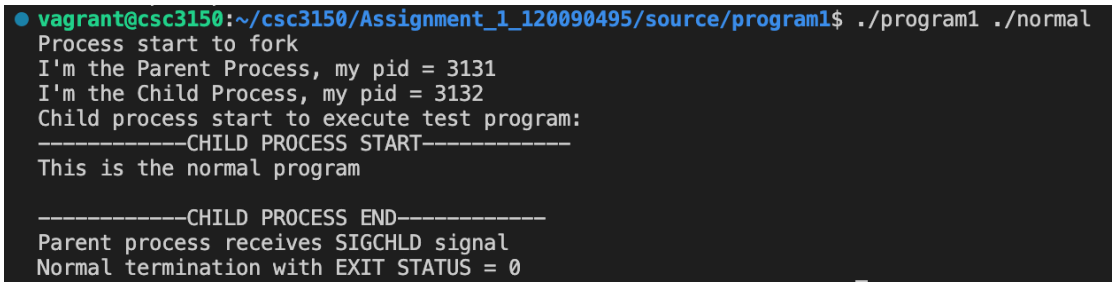
Task 1

Task 1 asks students to fork a child process to execute the test program. I use fork() function first in order to fork the child process. Then I get the return value of the fork() function. Different value refer to different situations. Then I establish the contact between arg[] and argv[] to make sure they are right. Using execve() to execute a test program. Parent process needs to wait until child process terminates. waitpid(-1, &status, WUNTRACED) is used to wait and receive a signal from the child process. WTERMSIG(status) function is used to recognize the returned signal and return a exact integer value which can help us do recognition. According to the returned signal, the parent process will print different results so that we can know the signal.

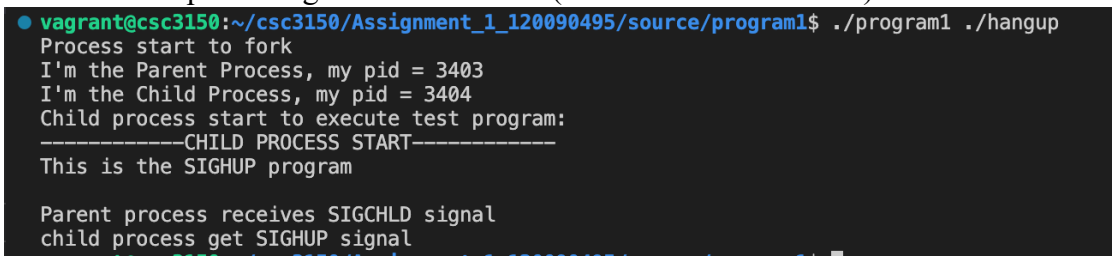
The environment of the program: Ubuntu 16.04 LTS Linux5.10.5

The screenshot of the output is as follows:

Demo output for normal termination:



Demo output for signaled termination (SIGCHLD for this case):



Demo output for stopped termination:

```
vagrant@csc3150:~/csc3150/Assignment_1_120090495/source/program1$ ./program1 ./stop
Process start to fork
I'm the Parent Process, my pid = 3429
I'm the Child Process, my pid = 3430
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGSTOP program

Parent process receives SIGCHLD signal
child process get SIGSTOP signal
```

Task 2

Task 2 asks me to initialize Kernel module first, after that I create kernel thread. Similar with program1, but use the kernel mode. I need to use the kernel related function to solve the problem. After that, I need to use export symbol in the /kernel/fork.c, /kernel/exit.c, /fs/exec.c, /fs/namei.c. Then re-compile the kernel.

Demo output for normal termination:

```
[ 3476.951317] [program2] : module_init Jiaron Zhang 120090495
[ 3476.951319] [program2] : module_init create kthread start
[ 3476.951320] [program2] : module_init kthread start
[ 3476.952920] [program2] : The child process has pid = 8627
[ 3476.952921] [program2] : This is the parent process, pid = 8625
[ 3476.952922] [program2] : child process
[ 3477.051944] [program2] : normal termination
[ 3477.051946] [program2] : child process terminated
[ 3477.051947] [program2] : The returned signal is 0
[ 3480.963937] [program2] : module_exit
root@csc3150:/home/vagrant/csc3150/Assignment_1_120090495/source/program2#
```

Demo output for signaled termination (SIGBUS for this case):

```
[ 5557.950947] [program2] : module_init Jiaron Zhang 120090495
[ 5557.950949] [program2] : module_init create kthread start
[ 5557.950949] [program2] : module_init kthread start
[ 5557.951194] [program2] : The child process has pid = 16064
[ 5557.951195] [program2] : This is the parent process, pid = 16063
[ 5557.951196] [program2] : child process
[ 5558.051778] [program2] : get SIGBUS signal
[ 5558.051780] [program2] : child process terminated
[ 5558.051781] [program2] : The returned signal is 7
[ 5562.118218] [program2] : module_exit
root@csc3150:/home/vagrant/csc3150/Assignment_1_120090495/source/program2#
```