CSC3150 Operating Systems

Report #1

Name: Song Chenghao

Student ID: 120090504

Date: 2022/10/09

The Chinese University of Hong Kong, Shenzhen

1.Design

1)program1

In user mode, fork a child process to execute the test program. Use function fork() and store the returned pid. Judge the pid to wake a parent or child process. When child process finish execution(execve()), the parent process will receive the SIGCHLD signal by waitpid() function. For different signals, print different messages about child process status.

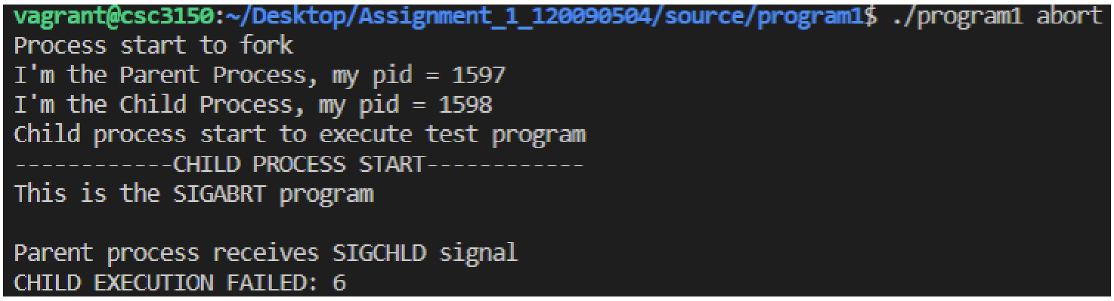
2)program2

In kernel mode, within my_fork(), fork a process to execute the test program. An important function in my_fork() is my_exec() that uses getname_kernel() to get filename from given path and execute the test program within do_execve(). Then, create a process within kernel_clone() in kernel and return a pid. Pass this pid of process to my_wait(), which get signals specially their status from do_wait(), print related messages in kernel log. The parent process will wait until child process terminates.

2. Development environment

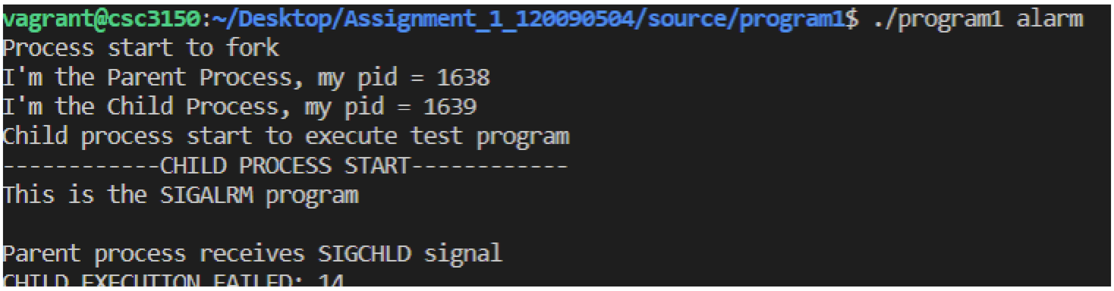
Connect VM to VSC, which has installed C language extend. In VSC, code has clang-format(external config) and termination. Download the kernel 5.10 form webpage. Copy compressed package to VM and decompress it. Copy the orignal config file(kernel 4.4) to decompressed directory. Clean the old config and make new config within orignal file. Build kernel image and modules to make sure it can be used correctly. Install kernal modules and kernel. After that, reboot to load ner kernel.

3.Program output



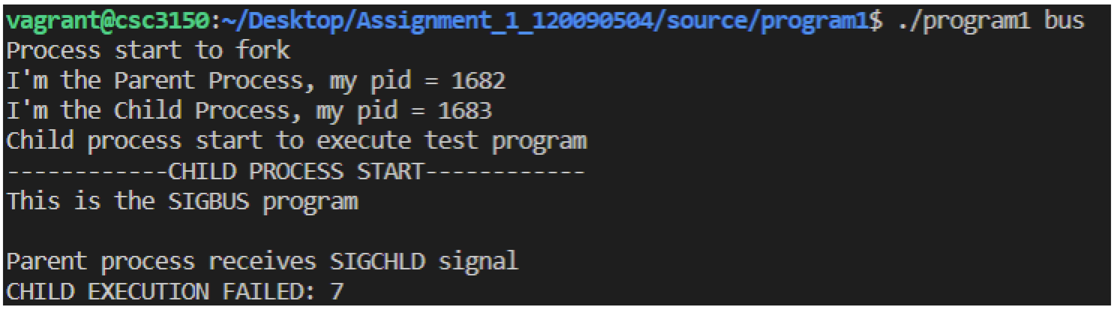
```
vagrant@csc3150:~/Desktop/Assignment_1_120090504/source/program1$ ./program1 abort
Process start to fork
I'm the Parent Process, my pid = 1597
I'm the Child Process, my pid = 1598
Child process start to execute test program
------------CHILD PROCESS START------------
This is the SIGABRT program

Parent process receives SIGCHLD signal
CHILD EXECUTION FAILED: 6
```



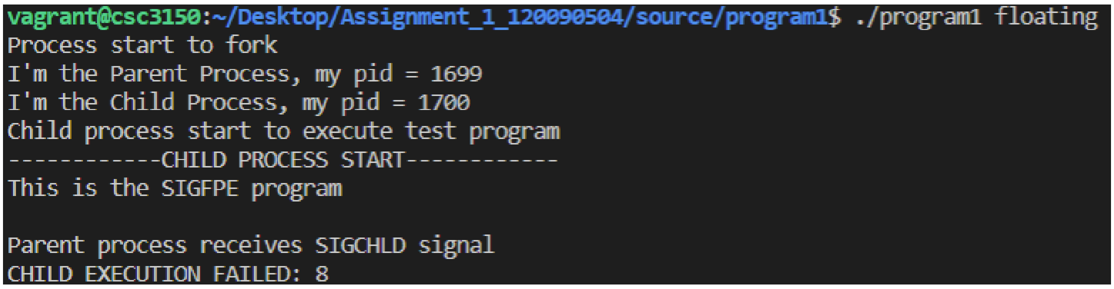
```
vagrant@csc3150:~/Desktop/Assignment_1_120090504/source/program1$ ./program1 alarm
Process start to fork
I'm the Parent Process, my pid = 1638
I'm the Child Process, my pid = 1639
Child process start to execute test program
------------CHILD PROCESS START------------
This is the SIGALRM program

Parent process receives SIGCHLD signal
CHILD EXECUTION FAILED: 14
```



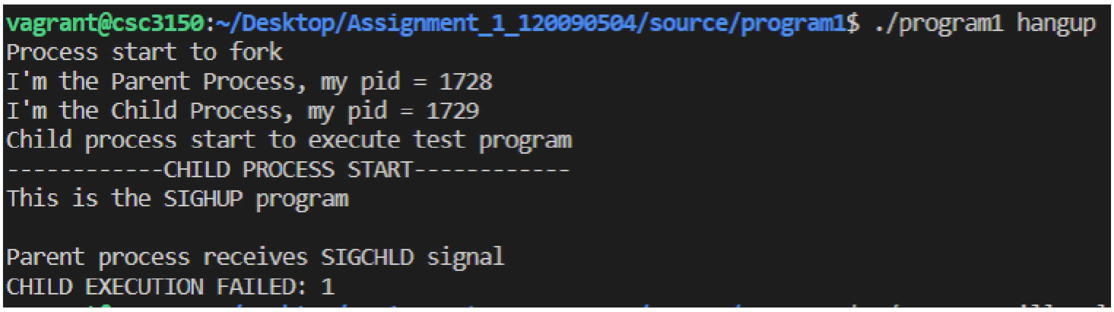
```
vagrant@csc3150:~/Desktop/Assignment_1_120090504/source/program1$ ./program1 bus
Process start to fork
I'm the Parent Process, my pid = 1682
I'm the Child Process, my pid = 1683
Child process start to execute test program
------------CHILD PROCESS START------------
This is the SIGBUS program

Parent process receives SIGCHLD signal
CHILD EXECUTION FAILED: 7
```



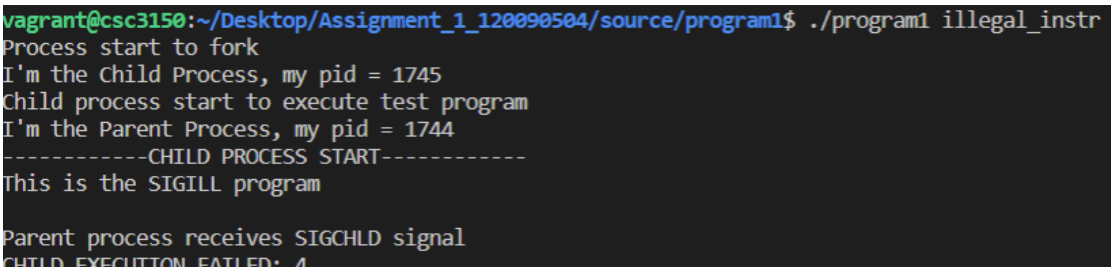
```
vagrant@csc3150:~/Desktop/Assignment_1_120090504/source/program1$ ./program1 floating
Process start to fork
I'm the Parent Process, my pid = 1699
I'm the Child Process, my pid = 1700
Child process start to execute test program
------------CHILD PROCESS START------------
This is the SIGFPE program

Parent process receives SIGCHLD signal
CHILD EXECUTION FAILED: 8
```



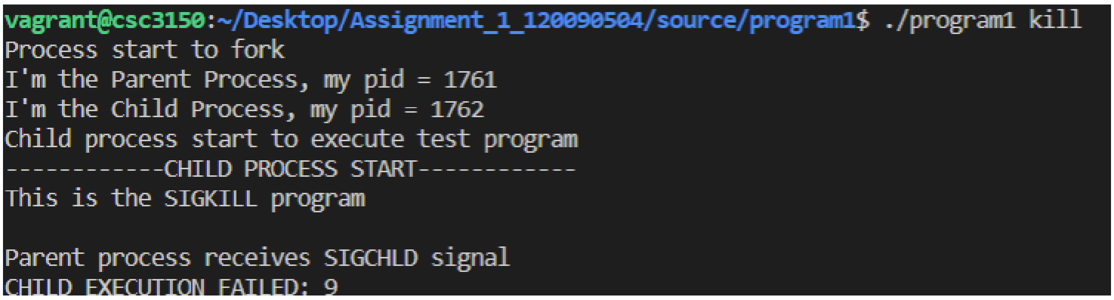
```
vagrant@csc3150:~/Desktop/Assignment_1_120090504/source/program1$ ./program1 hangup
Process start to fork
I'm the Parent Process, my pid = 1728
I'm the Child Process, my pid = 1729
Child process start to execute test program
------------CHILD PROCESS START------------
This is the SIGHUP program

Parent process receives SIGCHLD signal
CHILD EXECUTION FAILED: 1
```



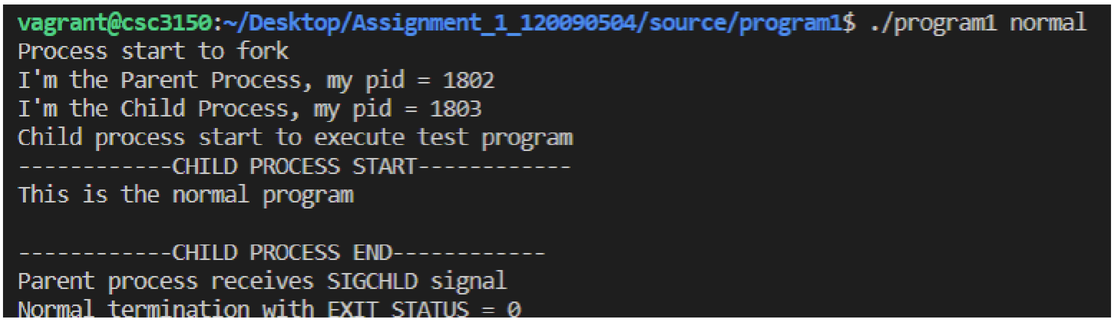
```
vagrant@csc3150:~/Desktop/Assignment_1_120090504/source/program1$ ./program1 illegal_instr
Process start to fork
I'm the Child Process, my pid = 1745
Child process start to execute test program
I'm the Parent Process, my pid = 1744
------------CHILD PROCESS START------------
This is the SIGILL program

Parent process receives SIGCHLD signal
CHILD EXECUTION FAILED: 4
```



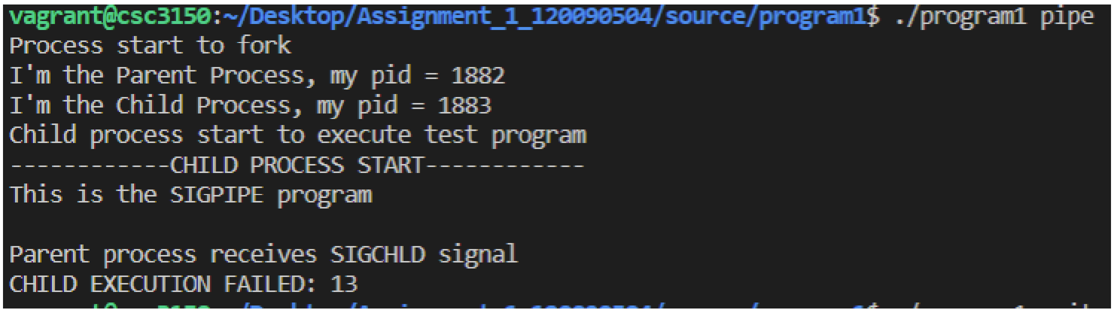
```
vagrant@csc3150:~/Desktop/Assignment_1_120090504/source/program1$ ./program1 kill
Process start to fork
I'm the Parent Process, my pid = 1761
I'm the Child Process, my pid = 1762
Child process start to execute test program
------------CHILD PROCESS START------------
This is the SIGKILL program

Parent process receives SIGCHLD signal
CHILD EXECUTION FAILED: 9
```



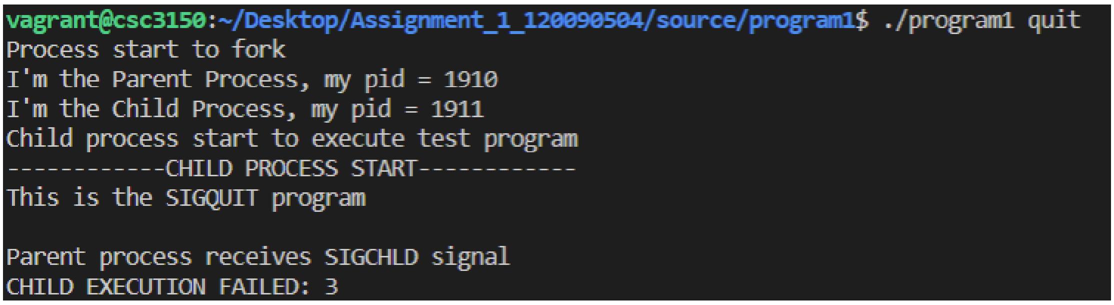
```
vagrant@csc3150:~/Desktop/Assignment_1_120090504/source/program1$ ./program1 normal
Process start to fork
I'm the Parent Process, my pid = 1802
I'm the Child Process, my pid = 1803
Child process start to execute test program
------------CHILD PROCESS START------------
This is the normal program

------------CHILD PROCESS END------------
Parent process receives SIGCHLD signal
Normal termination with EXIT STATUS = 0
```



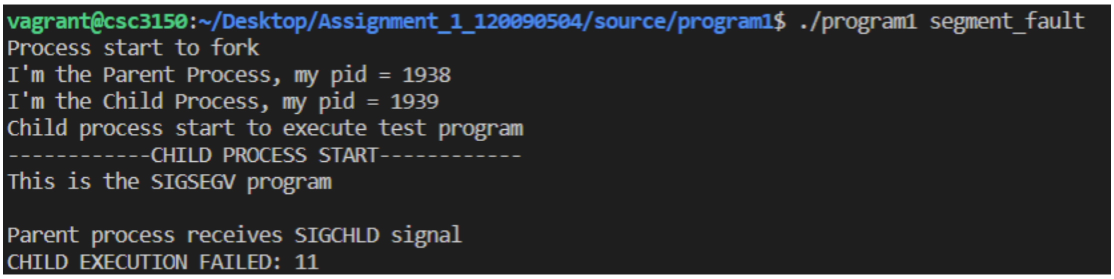
```
vagrant@csc3150:~/Desktop/Assignment_1_120090504/source/program1$ ./program1 pipe
Process start to fork
I'm the Parent Process, my pid = 1882
I'm the Child Process, my pid = 1883
Child process start to execute test program
------------CHILD PROCESS START------------
This is the SIGPIPE program

Parent process receives SIGCHLD signal
CHILD EXECUTION FAILED: 13
```



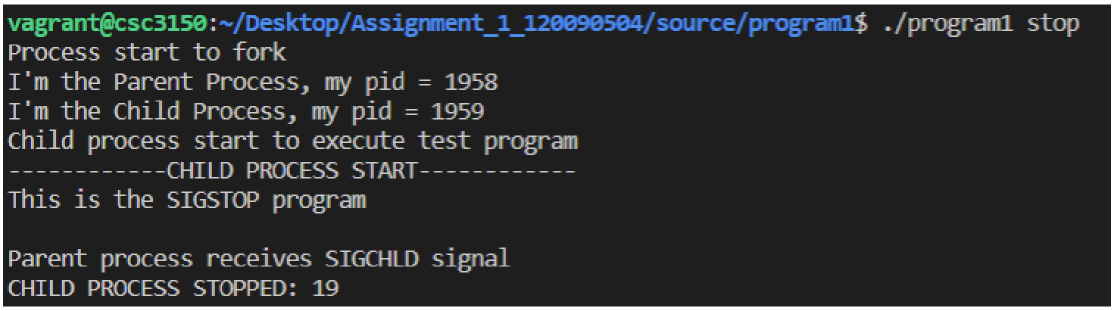
```
vagrant@csc3150:~/Desktop/Assignment_1_120090504/source/program1$ ./program1 quit
Process start to fork
I'm the Parent Process, my pid = 1910
I'm the Child Process, my pid = 1911
Child process start to execute test program
------------CHILD PROCESS START------------
This is the SIGQUIT program

Parent process receives SIGCHLD signal
CHILD EXECUTION FAILED: 3
```



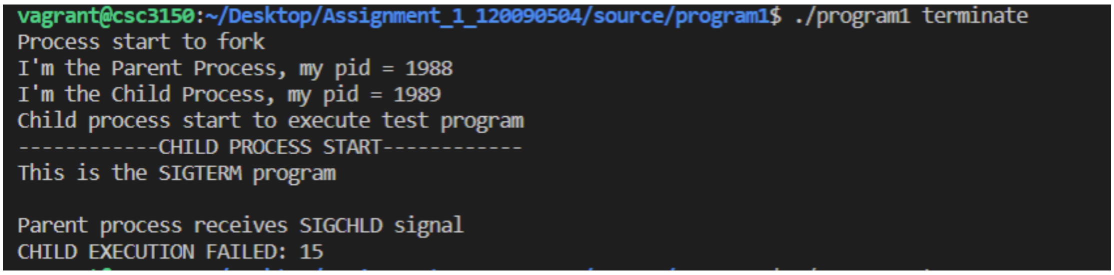
```
vagrant@csc3150:~/Desktop/Assignment_1_120090504/source/program1$ ./program1 segment_fault
Process start to fork
I'm the Parent Process, my pid = 1938
I'm the Child Process, my pid = 1939
Child process start to execute test program
------------CHILD PROCESS START------------
This is the SIGSEGV program

Parent process receives SIGCHLD signal
CHILD EXECUTION FAILED: 11
```



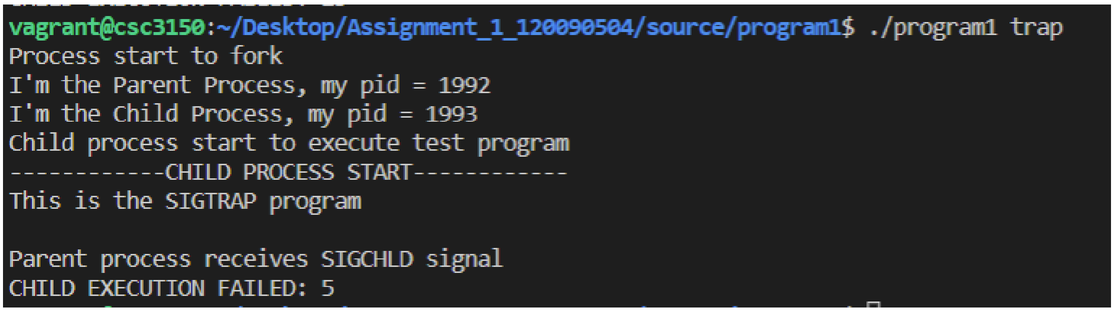
```
vagrant@csc3150:~/Desktop/Assignment_1_120090504/source/program1$ ./program1 stop
Process start to fork
I'm the Parent Process, my pid = 1958
I'm the Child Process, my pid = 1959
Child process start to execute test program
------------CHILD PROCESS START------------
This is the SIGSTOP program

Parent process receives SIGCHLD signal
CHILD PROCESS STOPPED: 19
```



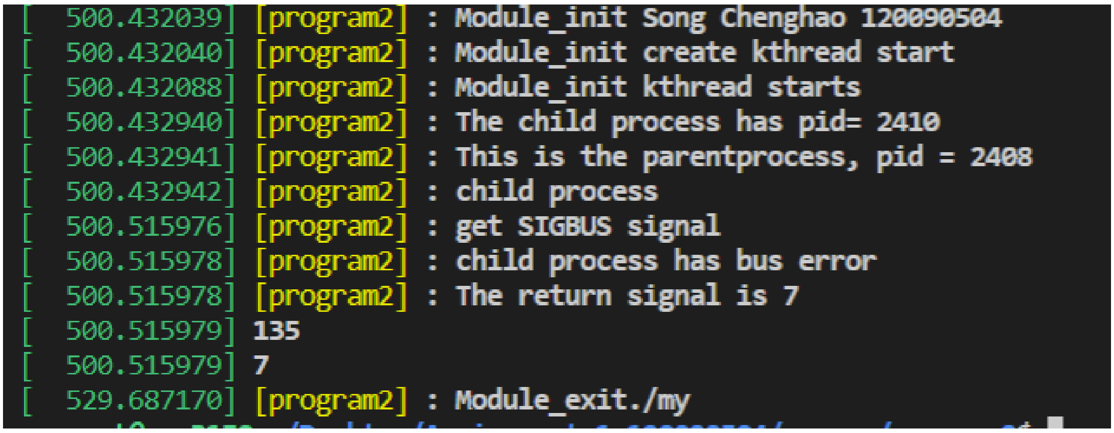
```
vagrant@csc3150:~/Desktop/Assignment_1_120090504/source/program1$ ./program1 terminate
Process start to fork
I'm the Parent Process, my pid = 1988
I'm the Child Process, my pid = 1989
Child process start to execute test program
------------CHILD PROCESS START------------
This is the SIGTERM program

Parent process receives SIGCHLD signal
CHILD EXECUTION FAILED: 15
```



```
vagrant@csc3150:~/Desktop/Assignment_1_120090504/source/program1$ ./program1 trap
Process start to fork
I'm the Parent Process, my pid = 1992
I'm the Child Process, my pid = 1993
Child process start to execute test program
------------CHILD PROCESS START------------
This is the SIGTRAP program

Parent process receives SIGCHLD signal
CHILD EXECUTION FAILED: 5
```



```
[  500.432039] [program2] : Module_init Song Chenghao 120090504
[  500.432040] [program2] : Module_init create kthread start
[  500.432088] [program2] : Module_init kthread starts
[  500.432940] [program2] : The child process has pid= 2410
[  500.432941] [program2] : This is the parentprocess, pid = 2408
[  500.432942] [program2] : child process
[  500.515976] [program2] : get SIGBUS signal
[  500.515978] [program2] : child process has bus error
[  500.515978] [program2] : The return signal is 7
[  500.515979] 135
[  500.515979] 7
[  529.687170] [program2] : Module_exit./my
```

4.Learn

I became more familiar with the commands in Linux and the c language. I learned the difference between kernel mode and user mode. I understood syscall and how to create processes and get signal information about processes in different modes.