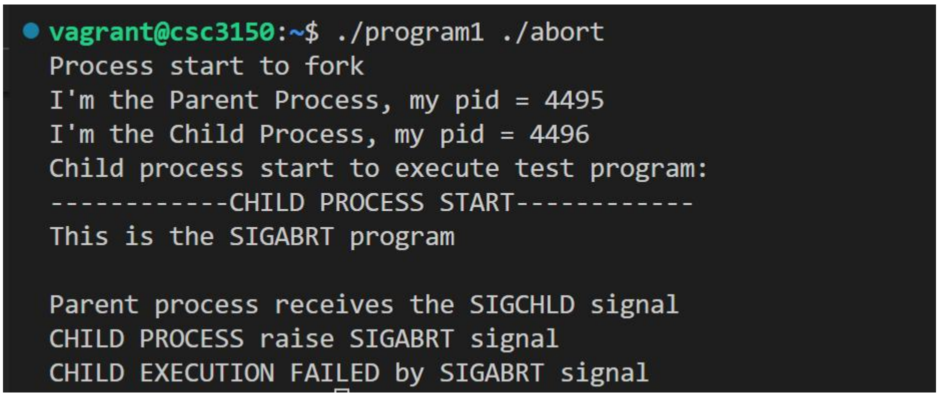
Report of CSC3150 assignment 1

曹潚 120090847

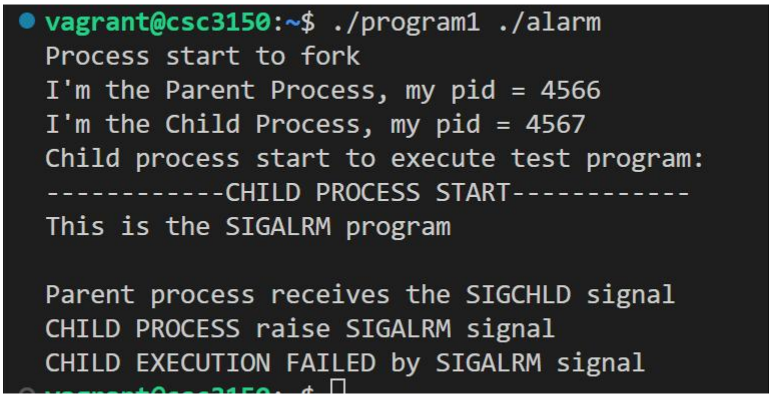
This assignment is mainly composed by task1 and task2. In task1, we need to fork a child process and identify the signal child process raises. We only need to complete task1 through user mode, so we just use fork() to raise the new process. Use waitpid() to catch the signal from child process.

Just follow the tutorial slides and write the code. Run the code , I found that each different signal will return different number(WTERMSIG(status)) to make the process execution failed. So I recorded the numbers and use if() and else if() to print different output sentences. The outputs are as follows.



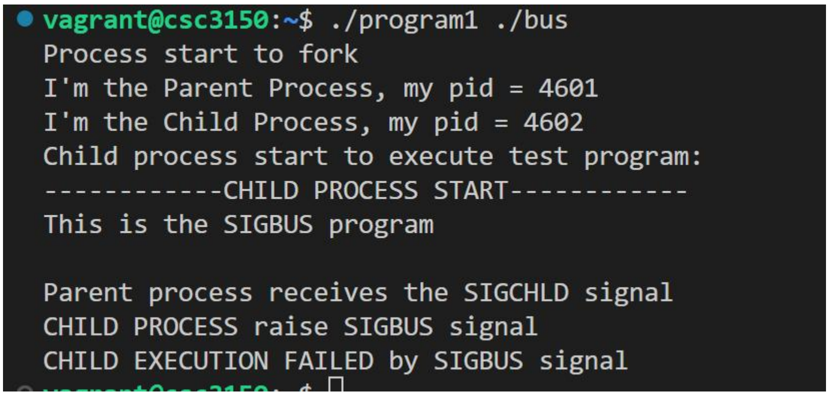
```
vagrant@csc3150:~$ ./program1 ./abort
Process start to fork
I'm the Parent Process, my pid = 4495
I'm the Child Process, my pid = 4496
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGABRT program

Parent process receives the SIGCHLD signal
CHILD PROCESS raise SIGABRT signal
CHILD EXECUTION FAILED by SIGABRT signal
```



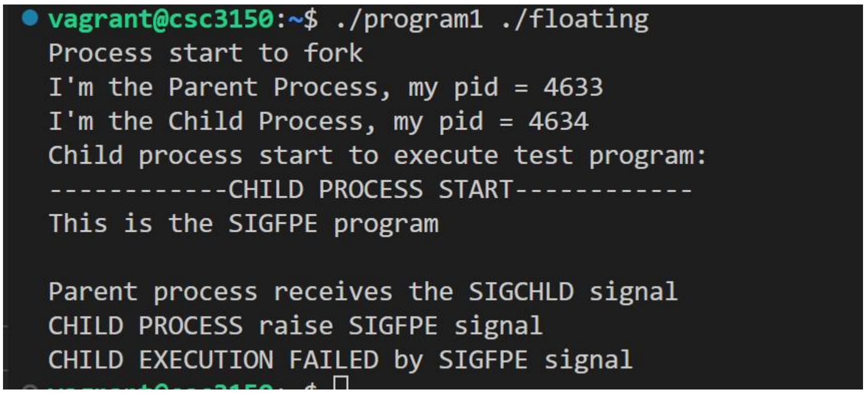
```
vagrant@csc3150:~$ ./program1 ./alarm
Process start to fork
I'm the Parent Process, my pid = 4566
I'm the Child Process, my pid = 4567
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGALRM program

Parent process receives the SIGCHLD signal
CHILD PROCESS raise SIGALRM signal
CHILD EXECUTION FAILED by SIGALRM signal
```



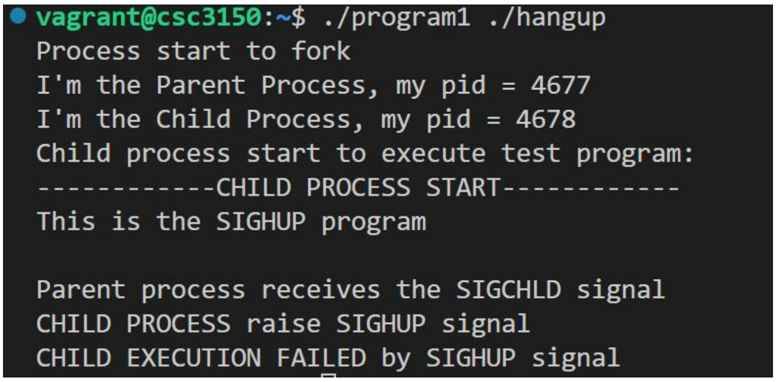
```
vagrant@csc3150:~$ ./program1 ./bus
Process start to fork
I'm the Parent Process, my pid = 4601
I'm the Child Process, my pid = 4602
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGBUS program

Parent process receives the SIGCHLD signal
CHILD PROCESS raise SIGBUS signal
CHILD EXECUTION FAILED by SIGBUS signal
```



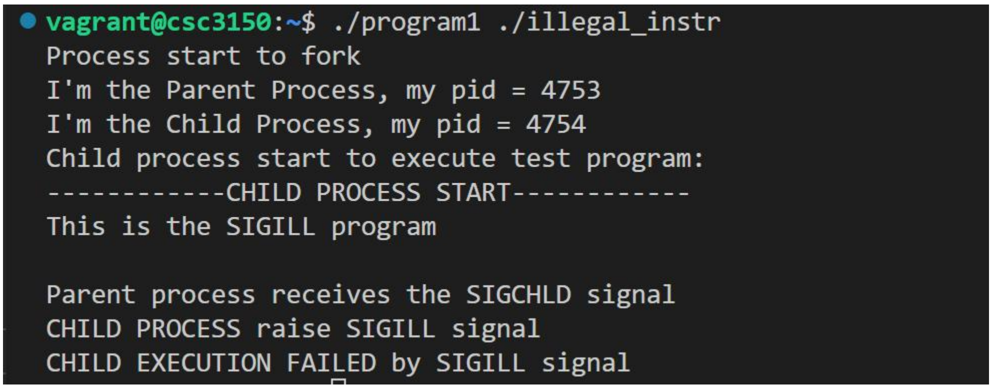
```
vagrant@csc3150:~$ ./program1 ./floating
Process start to fork
I'm the Parent Process, my pid = 4633
I'm the Child Process, my pid = 4634
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGFPE program

Parent process receives the SIGCHLD signal
CHILD PROCESS raise SIGFPE signal
CHILD EXECUTION FAILED by SIGFPE signal
```



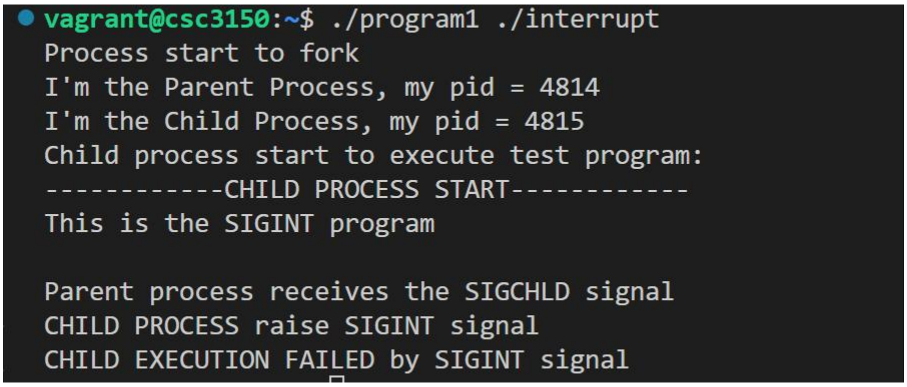
```
vagrant@csc3150:~$ ./program1 ./hangup
Process start to fork
I'm the Parent Process, my pid = 4677
I'm the Child Process, my pid = 4678
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGHUP program

Parent process receives the SIGCHLD signal
CHILD PROCESS raise SIGHUP signal
CHILD EXECUTION FAILED by SIGHUP signal
```



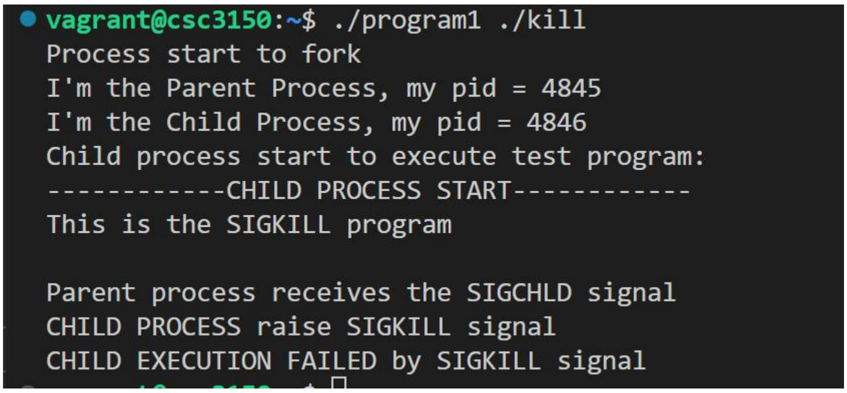
```
vagrant@csc3150:~$ ./program1 ./illegal_instr
Process start to fork
I'm the Parent Process, my pid = 4753
I'm the Child Process, my pid = 4754
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGILL program

Parent process receives the SIGCHLD signal
CHILD PROCESS raise SIGILL signal
CHILD EXECUTION FAILED by SIGILL signal
```



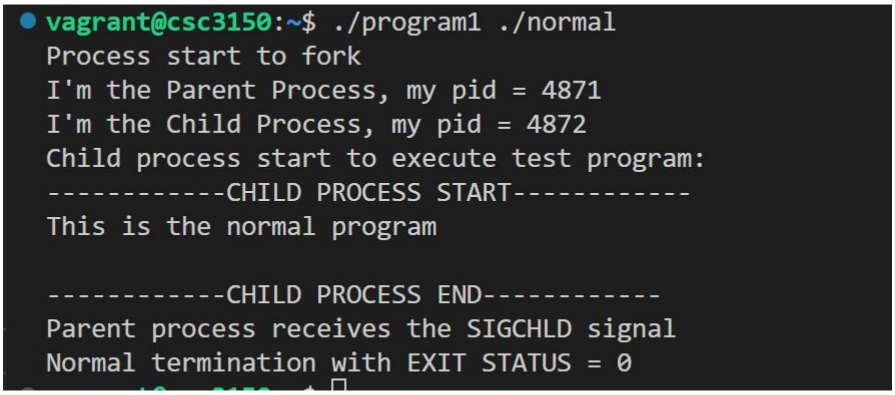
```
vagrant@csc3150:~$ ./program1 ./interrupt
Process start to fork
I'm the Parent Process, my pid = 4814
I'm the Child Process, my pid = 4815
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGINT program

Parent process receives the SIGCHLD signal
CHILD PROCESS raise SIGINT signal
CHILD EXECUTION FAILED by SIGINT signal
```



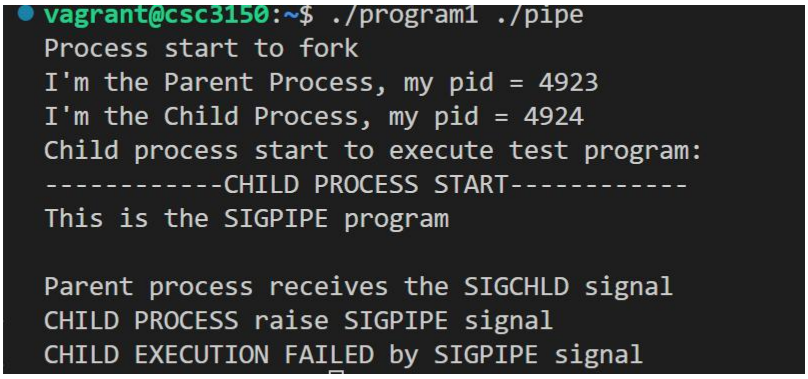
```
vagrant@csc3150:~$ ./program1 ./kill
Process start to fork
I'm the Parent Process, my pid = 4845
I'm the Child Process, my pid = 4846
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGKILL program

Parent process receives the SIGCHLD signal
CHILD PROCESS raise SIGKILL signal
CHILD EXECUTION FAILED by SIGKILL signal
```



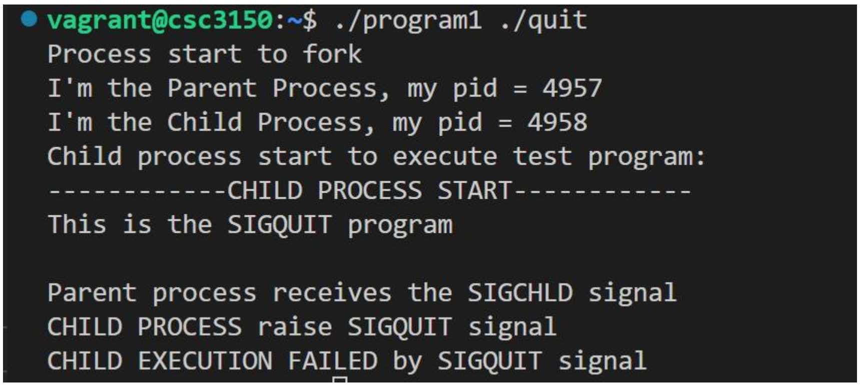
```
vagrant@csc3150:~$ ./program1 ./normal
Process start to fork
I'm the Parent Process, my pid = 4871
I'm the Child Process, my pid = 4872
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the normal program

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



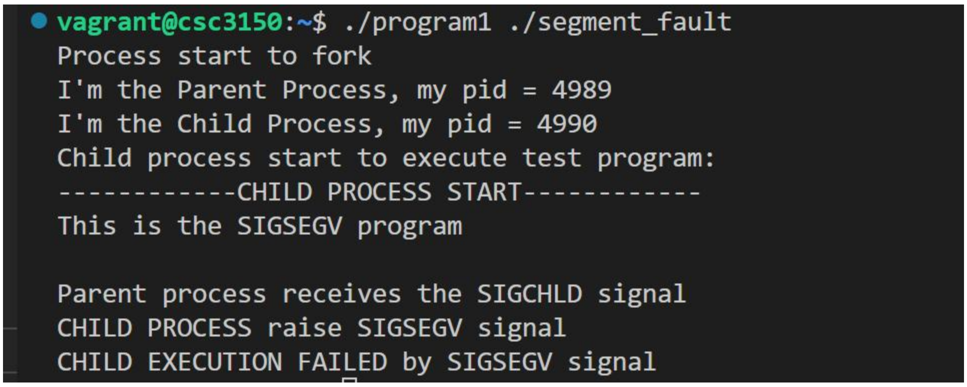
```
vagrant@csc3150:~$ ./program1 ./pipe
Process start to fork
I'm the Parent Process, my pid = 4923
I'm the Child Process, my pid = 4924
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGPIPE program

Parent process receives the SIGCHLD signal
CHILD PROCESS raise SIGPIPE signal
CHILD EXECUTION FAILED by SIGPIPE signal
```



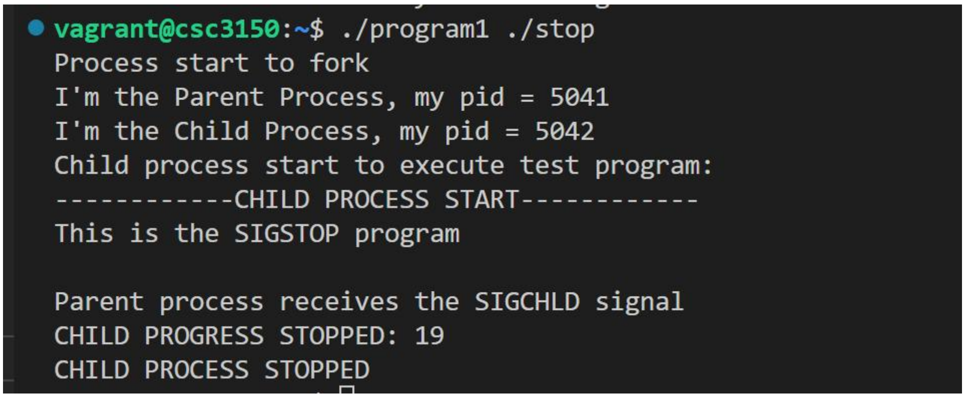
```
vagrant@csc3150:~$ ./program1 ./quit
Process start to fork
I'm the Parent Process, my pid = 4957
I'm the Child Process, my pid = 4958
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGQUIT program

Parent process receives the SIGCHLD signal
CHILD PROCESS raise SIGQUIT signal
CHILD EXECUTION FAILED by SIGQUIT signal
```



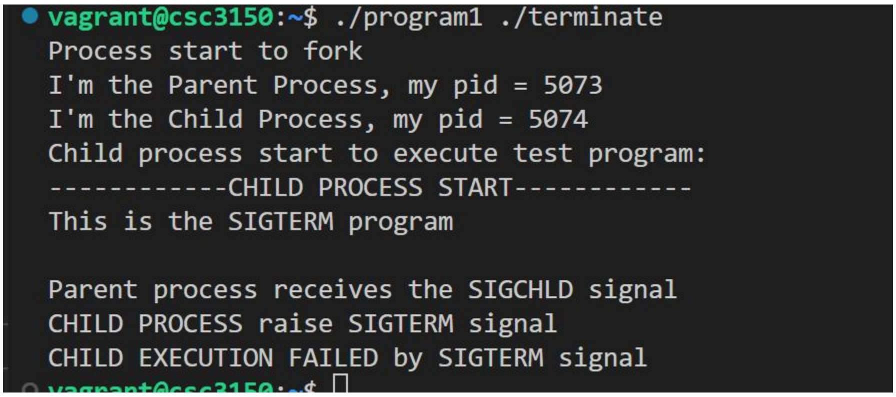
```
vagrant@csc3150:~$ ./program1 ./segment_fault
Process start to fork
I'm the Parent Process, my pid = 4989
I'm the Child Process, my pid = 4990
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGSEGV program

Parent process receives the SIGCHLD signal
CHILD PROCESS raise SIGSEGV signal
CHILD EXECUTION FAILED by SIGSEGV signal
```



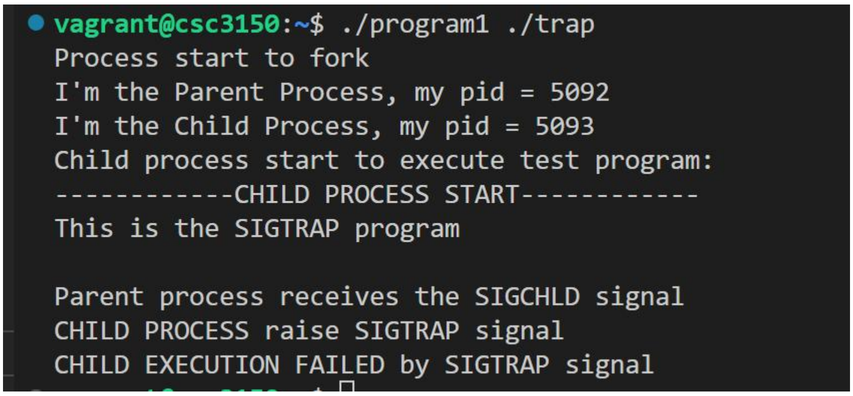
```
vagrant@csc3150:~$ ./program1 ./stop
Process start to fork
I'm the Parent Process, my pid = 5041
I'm the Child Process, my pid = 5042
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGSTOP program

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



```
vagrant@csc3150:~$ ./program1 ./terminate
Process start to fork
I'm the Parent Process, my pid = 5073
I'm the Child Process, my pid = 5074
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGTERM program

Parent process receives the SIGCHLD signal
CHILD PROCESS raise SIGTERM signal
CHILD EXECUTION FAILED by SIGTERM signal
vagrant@csc3150:~$
```



```
vagrant@csc3150:~$ ./program1 ./trap
Process start to fork
I'm the Parent Process, my pid = 5092
I'm the Child Process, my pid = 5093
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGTRAP program

Parent process receives the SIGCHLD signal
CHILD PROCESS raise SIGTRAP signal
CHILD EXECUTION FAILED by SIGTRAP signal
```

For environment setup, first download virtual box and vagrant source. After installing, we need to reboot our machine. In virtual box, click system icon to enlarge the memory for compiling the kernel. Start powershell by using administrator authority, start the virtual machine. Open vscode and download ssh remote windows. In remote windows, first download kernel source. Then we can decompress the source to a folder we make. Then copy the old config from /boot to the new linux folder. Make menuconfig and save the .config. After that, run the command make -j$(nproc), the kernel will be set up. Finally, use command make module_install and make install. The kernel will be compiled successfully.