Report on Assignment 1

Jiang Yige 121090233

Task 1

How I set up my development environment

Task1 is completed on **MacOS (ARM64/M1/M2)**

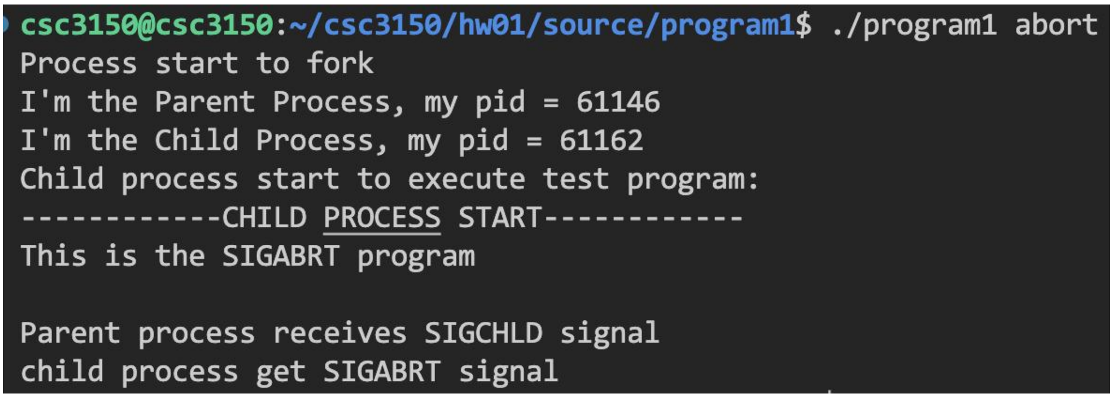
1. Download and install UTM and set up the VM.
2. Set up Remote SSH plugin in VS Code.
3. Update and Upgrade sources and Install build-essential.

How I design my program

1. Learn the example code on tutorial.
2. Use *fork()* to create child process.
3. Use the return value of *fork()* to detect whether the program is in the parent process or child process.
4. Use *getpid()* and *getppid()* to get current pid and parent pid.
5. Use *execve()* to execute the child process.
6. Use *waitpid()* to hold the parent process until receiving a signal from the child process.
7. Use *WIFEXITED() WIFSTOPPED() WIFSIGNALED()* to detect the received signal.
8. *Char* table[16]* is an array of strings, and *table[i]* refers to a signal with i-th signal type.

My program output

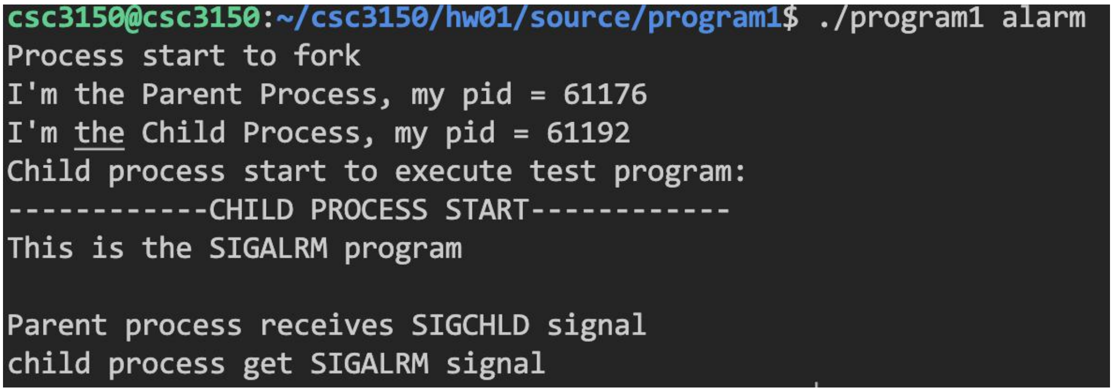
Abort



```
csc3150@csc3150:~/csc3150/hw01/source/program1$ ./program1 abort
Process start to fork
I'm the Parent Process, my pid = 61146
I'm the Child Process, my pid = 61162
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGABRT program

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

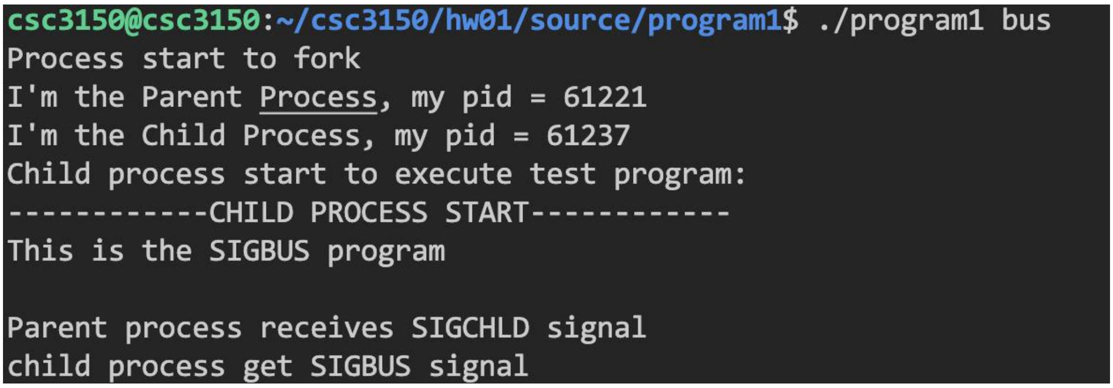
Alarm



```
csc3150@csc3150:~/csc3150/hw01/source/program1$ ./program1 alarm
Process start to fork
I'm the Parent Process, my pid = 61176
I'm the Child Process, my pid = 61192
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGALRM program

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

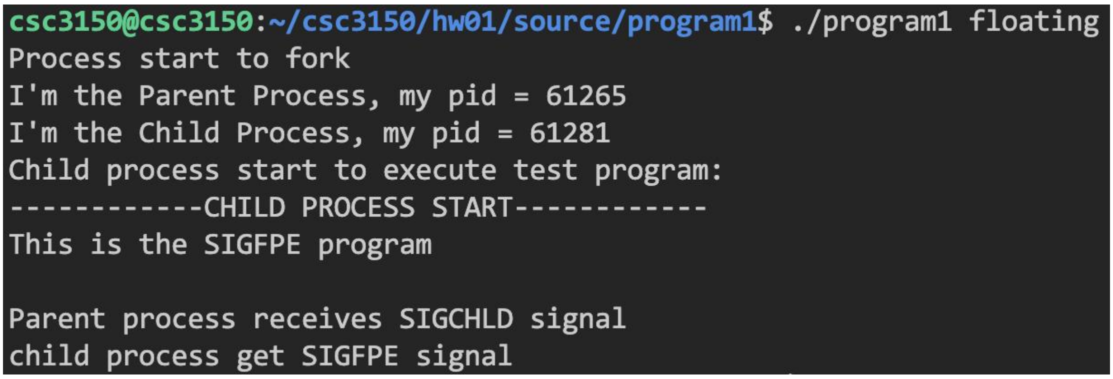
Bus



```
csc3150@csc3150:~/csc3150/hw01/source/program1$ ./program1 bus
Process start to fork
I'm the Parent Process, my pid = 61221
I'm the Child Process, my pid = 61237
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGBUS program

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

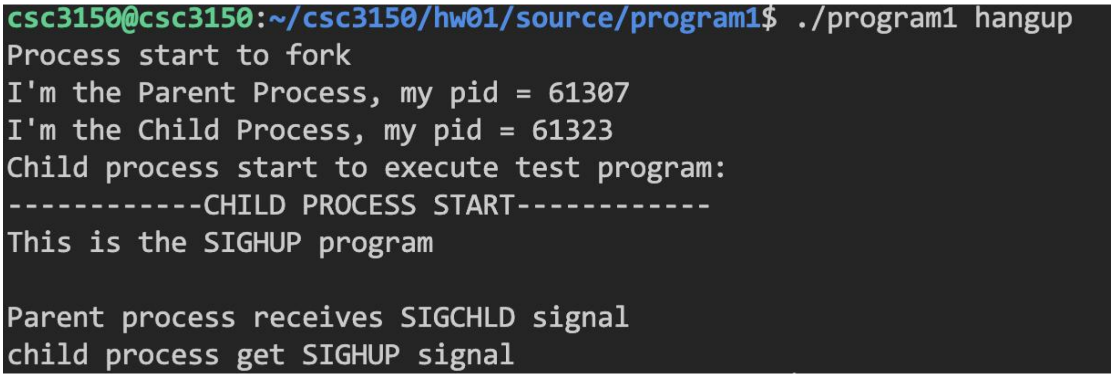
Floating



```
csc3150@csc3150:~/csc3150/hw01/source/program1$ ./program1 floating
Process start to fork
I'm the Parent Process, my pid = 61265
I'm the Child Process, my pid = 61281
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGFPE program

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

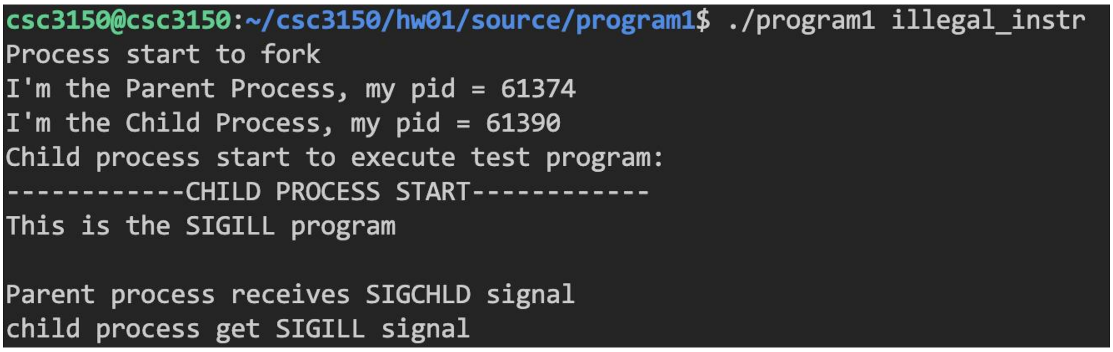
Hangup



```
csc3150@csc3150:~/csc3150/hw01/source/program1$ ./program1 hangup
Process start to fork
I'm the Parent Process, my pid = 61307
I'm the Child Process, my pid = 61323
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGHUP program

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

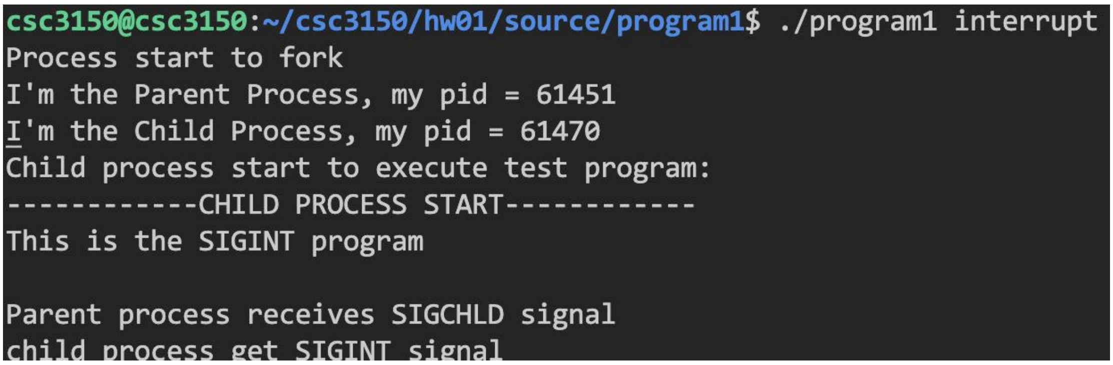
Illegal_instr



```
csc3150@csc3150:~/csc3150/hw01/source/program1$ ./program1 illegal_instr
Process start to fork
I'm the Parent Process, my pid = 61374
I'm the Child Process, my pid = 61390
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGILL program

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

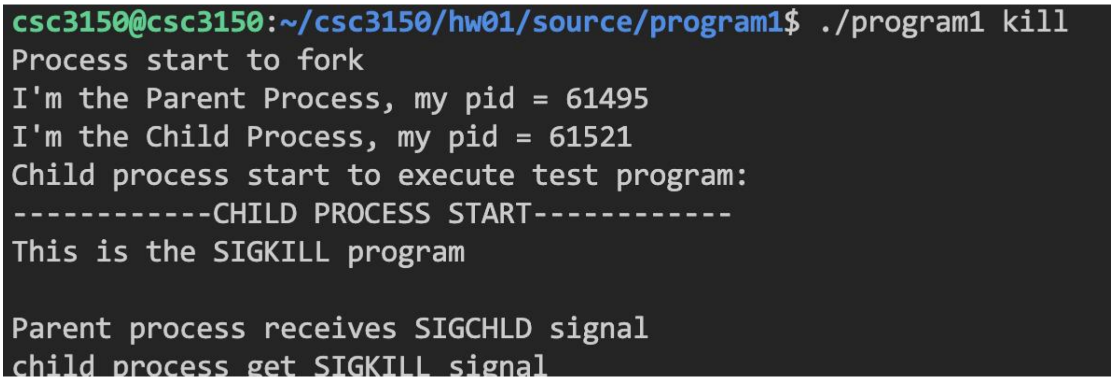
Interrupt



```
csc3150@csc3150:~/csc3150/hw01/source/program1$ ./program1 interrupt
Process start to fork
I'm the Parent Process, my pid = 61451
I'm the Child Process, my pid = 61470
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGINT program

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

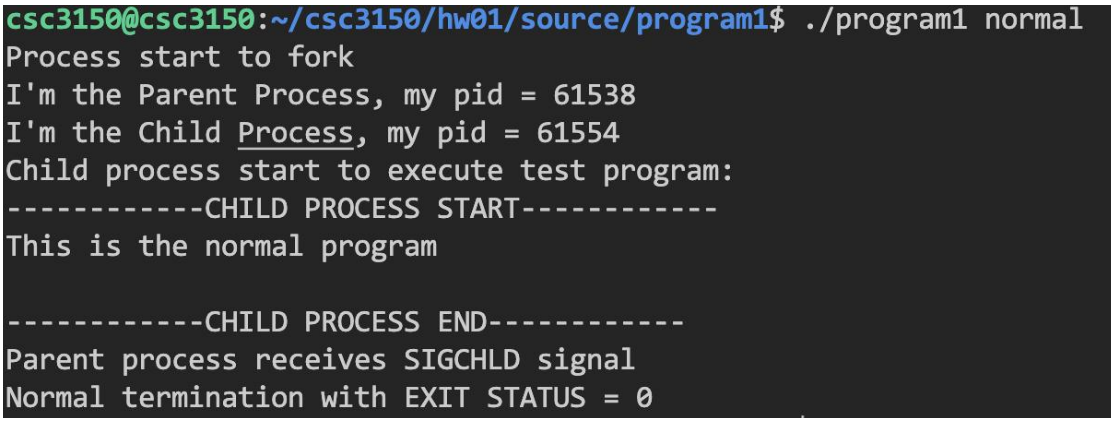
Kill



```
csc3150@csc3150:~/csc3150/hw01/source/program1$ ./program1 kill
Process start to fork
I'm the Parent Process, my pid = 61495
I'm the Child Process, my pid = 61521
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGKILL program

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

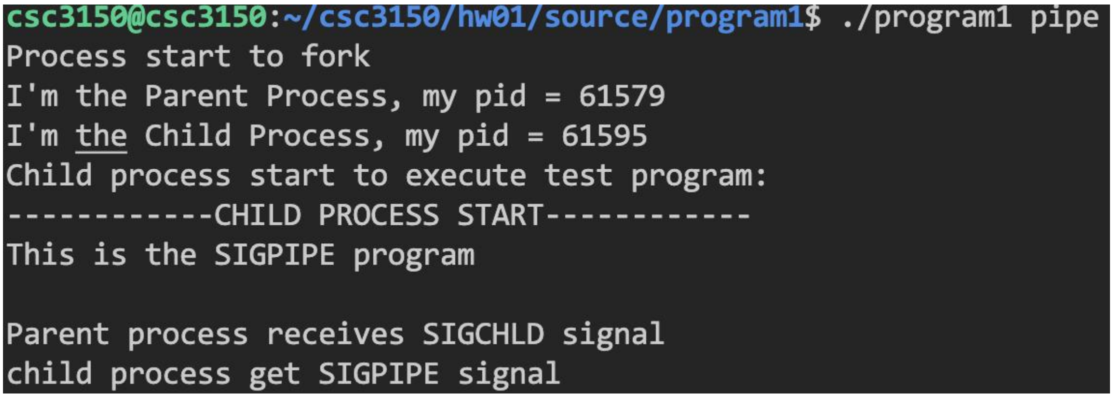
Normal



```
csc3150@csc3150:~/csc3150/hw01/source/program1$ ./program1 normal
Process start to fork
I'm the Parent Process, my pid = 61538
I'm the Child Process, my pid = 61554
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
```

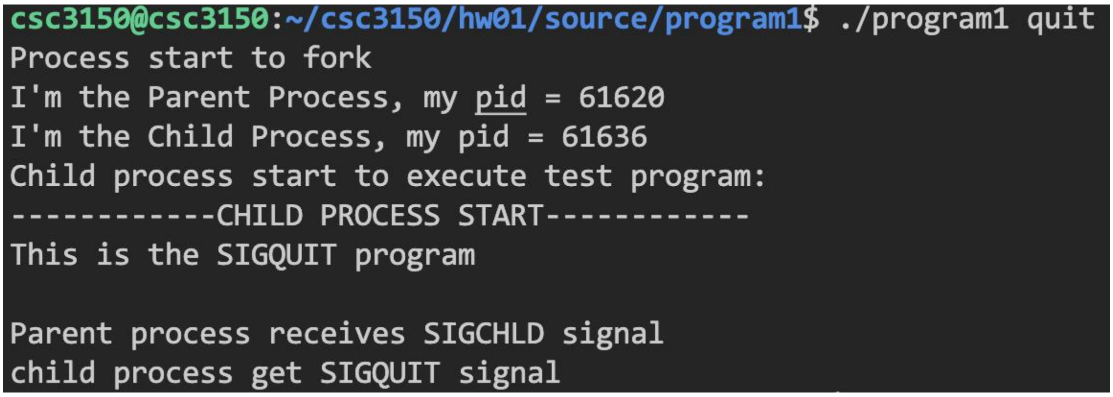
Pipe



```
csc3150@csc3150:~/csc3150/hw01/source/program1$ ./program1 pipe
Process start to fork
I'm the Parent Process, my pid = 61579
I'm the Child Process, my pid = 61595
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGPIPE program

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

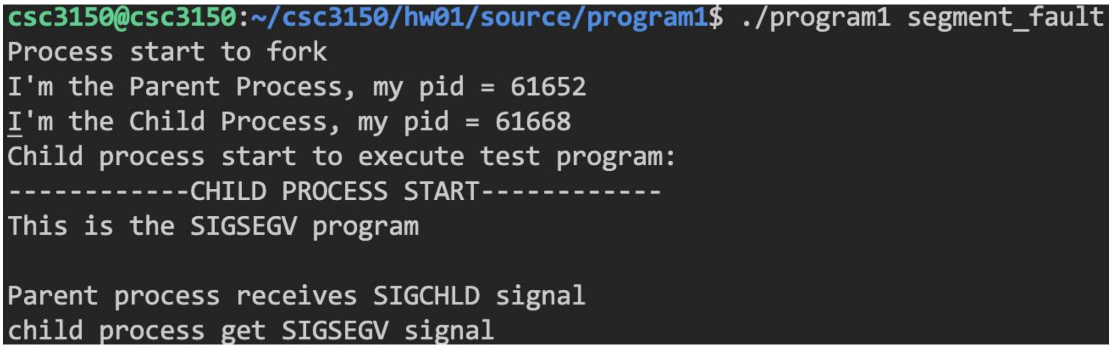
Quit



```
csc3150@csc3150:~/csc3150/hw01/source/program1$ ./program1 quit
Process start to fork
I'm the Parent Process, my pid = 61620
I'm the Child Process, my pid = 61636
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGQUIT program

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

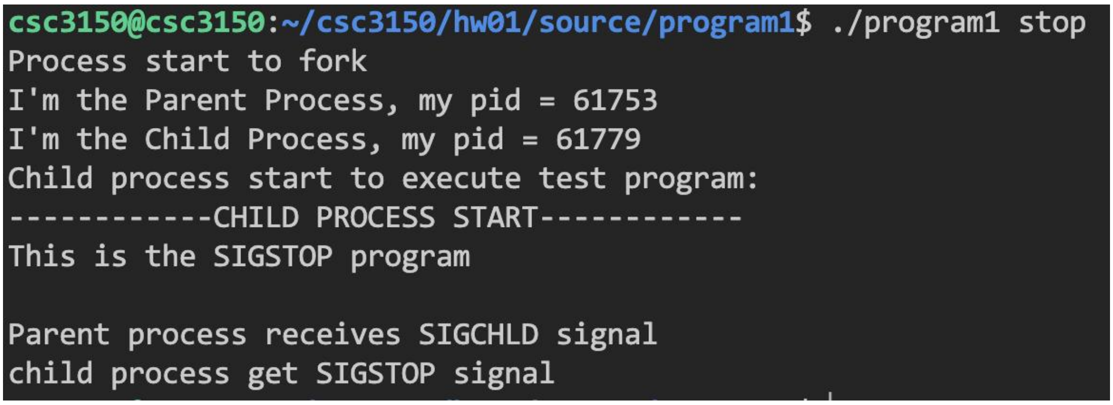
Segment_fault



```
csc3150@csc3150:~/csc3150/hw01/source/program1$ ./program1 segment_fault
Process start to fork
I'm the Parent Process, my pid = 61652
I'm the Child Process, my pid = 61668
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGSEGV program

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

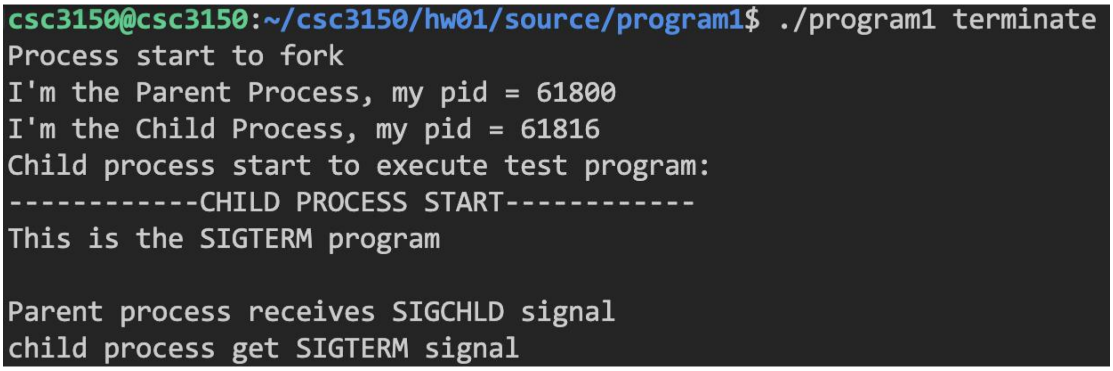
Stop



```
csc3150@csc3150:~/csc3150/hw01/source/program1$ ./program1 stop
Process start to fork
I'm the Parent Process, my pid = 61753
I'm the Child Process, my pid = 61779
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
```

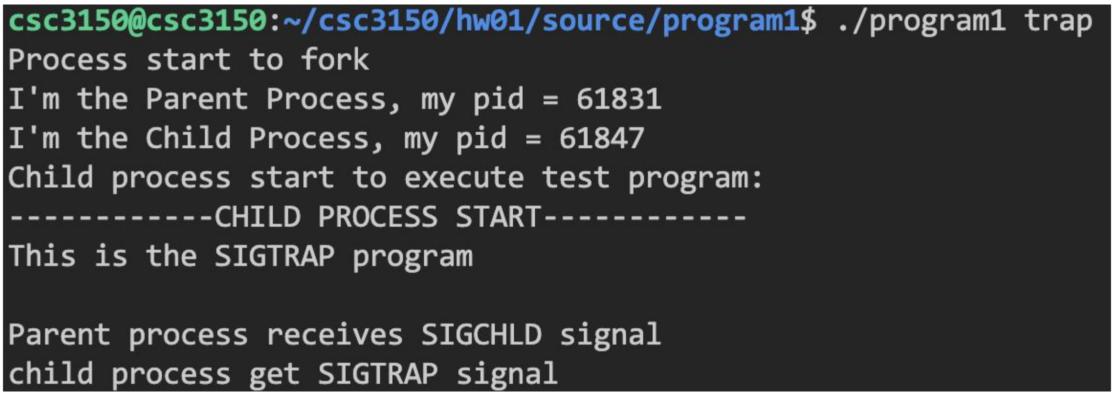
Terminate



```
csc3150@csc3150:~/csc3150/hw01/source/program1$ ./program1 terminate
Process start to fork
I'm the Parent Process, my pid = 61800
I'm the Child Process, my pid = 61816
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGTERM program

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

Trap



```
csc3150@csc3150:~/csc3150/hw01/source/program1$ ./program1 trap
Process start to fork
I'm the Parent Process, my pid = 61831
I'm the Child Process, my pid = 61847
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGTRAP program

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

What I learned

1. How a process is created on user mode.
2. How to detect whether the program is in the parent process or child process by fork().
3. How to get current pid and parent pid.
4. Receive and identify signals from the child process.
5. Use *WIFSTOPPED()* to detect whether the child process is stopped normally.

Task 2

How I set up my development environment

Task2 is completed on **Windows (AMD/Intel x86)**

Steps:

1. Set up the VM.

2. Install Dependency and development tools

3. Download the kernel with *5.10.x*.

4. Export the functions used in program2 from the head files.

   (kernel_clone(), getname_kernel, do_wait(), do_execve())

5. Compile kernel step by step by Tutorial.

Problem met:

On MacOS (ARM64/M1/M2):

1. The initial kernel version is *5.15.47*. Can't find the proper kernel sources such as *fork.c, exit.c, namei.c and exec.c.*
2. When trying to compile the kernel with *5.10.x*, can't *make dzImage -j*.
3. When trying to execute make modules -j , the VM crashed.

On Windows (AMD/Intel x86)

1. When compiling the kernel with *5.1*0.x, after executing *make modules -j* in root mode, the VM can't reboot normally. When start-up the VM, an error was raised: *7 urandom warning(s) missed due to ratelimiting*. The problem has been solved by allocating more

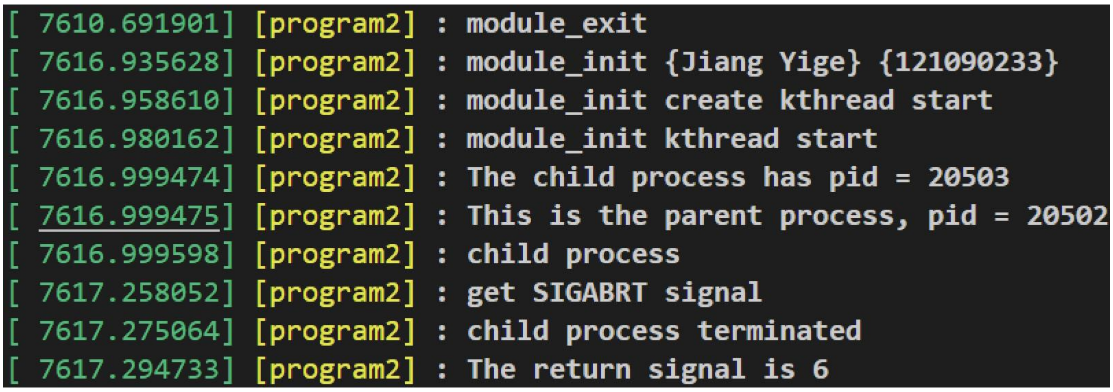
memory and CPUs to the VM.

How I design my program

1. Extern the functions mentioned in Hints.
2. Implement *my_fork(), my_wait(), my_exec, my_wait*;
3. In function *program2_init()*, use *kthread_create* to invoke *my_fork()*
4. In function *my_fork()*, define a *struct kernel_clone_args kargs* according to the tutorial, and use *kargs* to invoke *kernel_clone*.
5. In function *my_fork()*, use *kernel_clone* to fork a new process and get the child process' pid.
6. In function *my_fork()*, use *current* -> *pid* to get pid.
7. Use *my_wait()* to wait until the child process terminates and return a signal.
8. I implement function *getsignal()* to get the signal type ranged in [0, 19].
9. *Char* table[16]* is an array of strings, and *table[i]* refers to a signal with i-th signal type.

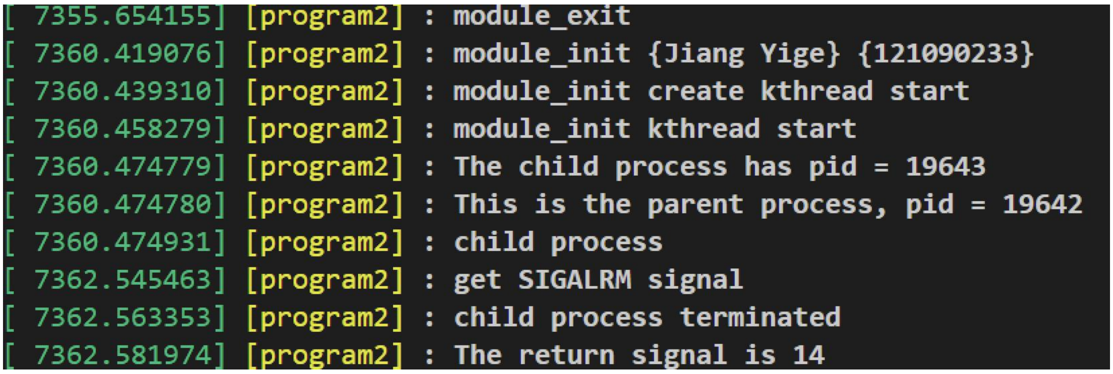
My program output

Abort



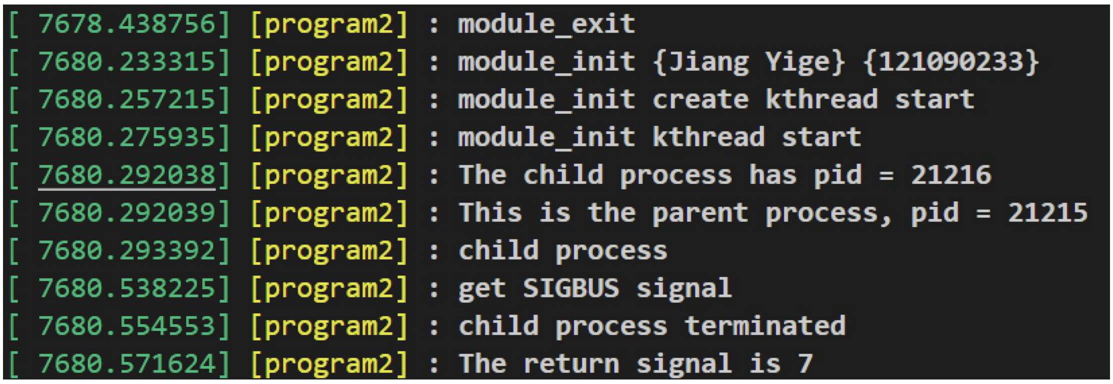
```
[ 7610.691901] [program2] : module_exit
[ 7616.935628] [program2] : module_init {Jiang Yige} {121090233}
[ 7616.958610] [program2] : module_init create kthread start
[ 7616.980162] [program2] : module_init kthread start
[ 7616.999474] [program2] : The child process has pid = 20503
[ 7616.999475] [program2] : This is the parent process, pid = 20502
[ 7616.999598] [program2] : child process
[ 7617.258052] [program2] : get SIGABRT signal
[ 7617.275064] [program2] : child process terminated
[ 7617.294733] [program2] : The return signal is 6
```

Alarm



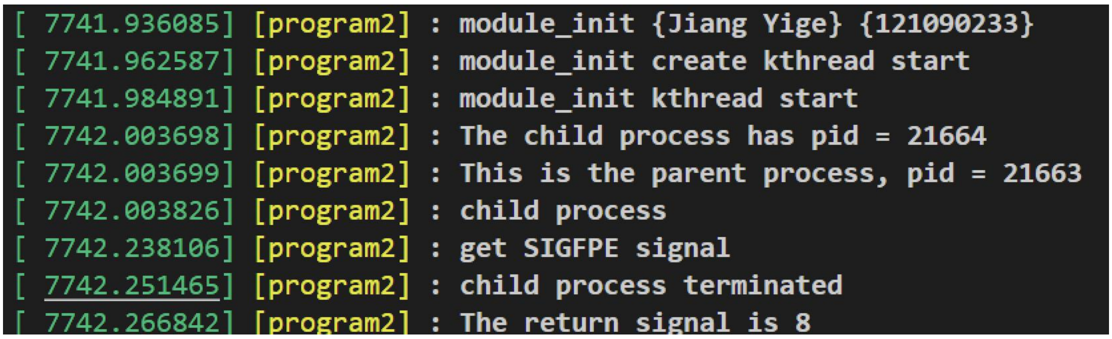
```
[ 7355.654155] [program2] : module_exit
[ 7360.419076] [program2] : module_init {Jiang Yige} {121090233}
[ 7360.439310] [program2] : module_init create kthread start
[ 7360.458279] [program2] : module_init kthread start
[ 7360.474779] [program2] : The child process has pid = 19643
[ 7360.474780] [program2] : This is the parent process, pid = 19642
[ 7360.474931] [program2] : child process
[ 7362.545463] [program2] : get SIGALRM signal
[ 7362.563353] [program2] : child process terminated
[ 7362.581974] [program2] : The return signal is 14
```

Bus



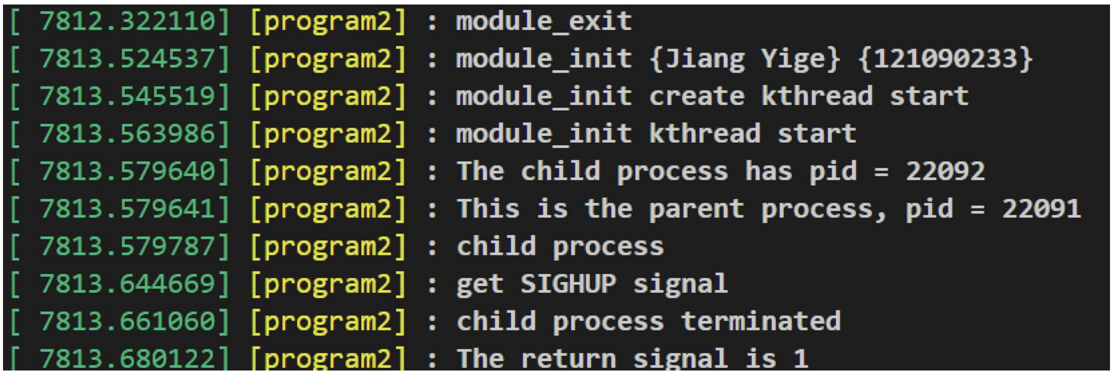
```
[ 7678.438756] [program2] : module_exit
[ 7680.233315] [program2] : module_init {Jiang Yige} {121090233}
[ 7680.257215] [program2] : module_init create kthread start
[ 7680.275935] [program2] : module_init kthread start
[ 7680.292038] [program2] : The child process has pid = 21216
[ 7680.292039] [program2] : This is the parent process, pid = 21215
[ 7680.293392] [program2] : child process
[ 7680.538225] [program2] : get SIGBUS signal
[ 7680.554553] [program2] : child process terminated
[ 7680.571624] [program2] : The return signal is 7
```

Floating



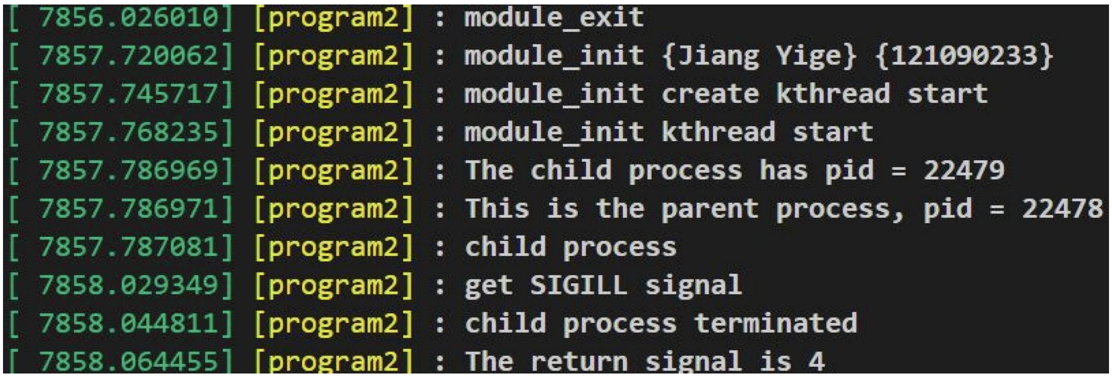
```
[ 7741.936085] [program2] : module_init {Jiang Yige} {121090233}
[ 7741.962587] [program2] : module_init create kthread start
[ 7741.984891] [program2] : module_init kthread start
[ 7742.003698] [program2] : The child process has pid = 21664
[ 7742.003699] [program2] : This is the parent process, pid = 21663
[ 7742.003826] [program2] : child process
[ 7742.238106] [program2] : get SIGFPE signal
[ 7742.251465] [program2] : child process terminated
[ 7742.266842] [program2] : The return signal is 8
```

Hangup



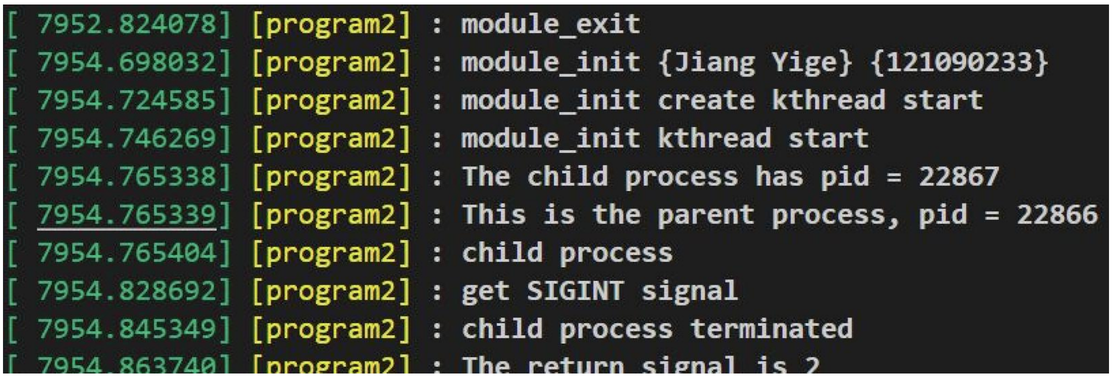
```
[ 7812.322110] [program2] : module_exit
[ 7813.524537] [program2] : module_init {Jiang Yige} {121090233}
[ 7813.545519] [program2] : module_init create kthread start
[ 7813.563986] [program2] : module_init kthread start
[ 7813.579640] [program2] : The child process has pid = 22092
[ 7813.579641] [program2] : This is the parent process, pid = 22091
[ 7813.579787] [program2] : child process
[ 7813.644669] [program2] : get SIGHUP signal
[ 7813.661060] [program2] : child process terminated
[ 7813.680122] [program2] : The return signal is 1
```

Illegal_instr



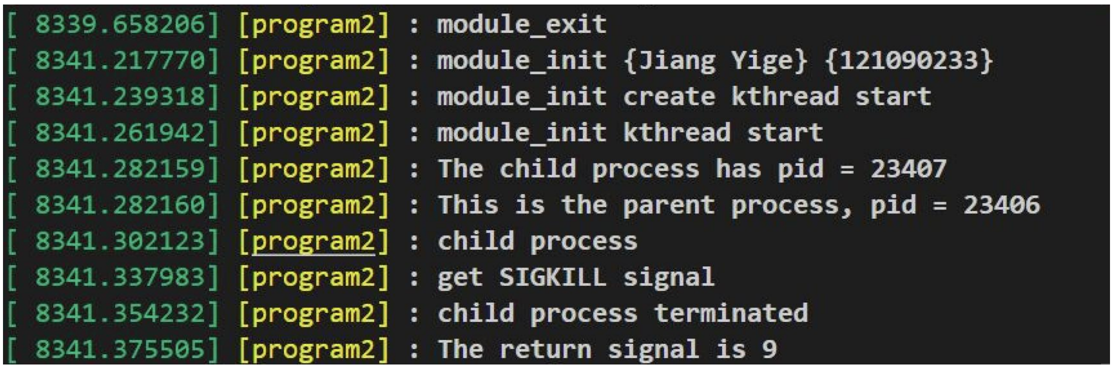
```
[ 7856.026010] [program2] : module_exit
[ 7857.720062] [program2] : module_init {Jiang Yige} {121090233}
[ 7857.745717] [program2] : module_init create kthread start
[ 7857.768235] [program2] : module_init kthread start
[ 7857.786969] [program2] : The child process has pid = 22479
[ 7857.786971] [program2] : This is the parent process, pid = 22478
[ 7857.787081] [program2] : child process
[ 7858.029349] [program2] : get SIGILL signal
[ 7858.044811] [program2] : child process terminated
[ 7858.064455] [program2] : The return signal is 4
```

Interrupt



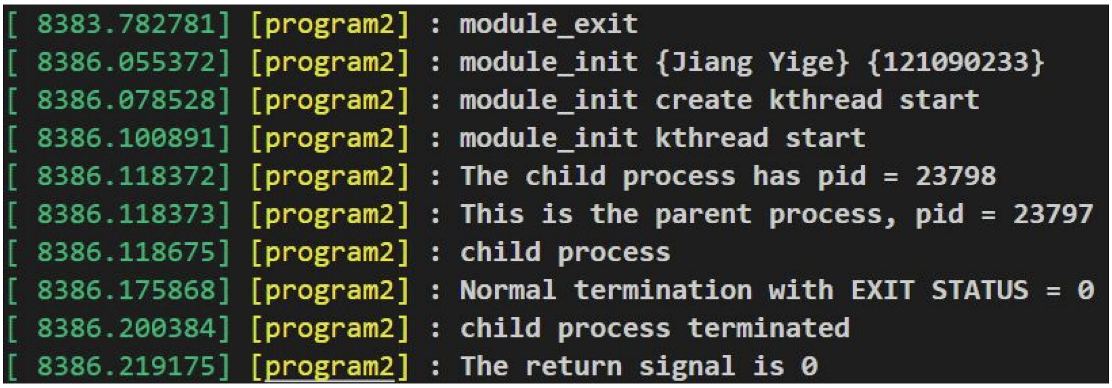
```
[ 7952.824078] [program2] : module_exit
[ 7954.698032] [program2] : module_init {Jiang Yige} {121090233}
[ 7954.724585] [program2] : module_init create kthread start
[ 7954.746269] [program2] : module_init kthread start
[ 7954.765338] [program2] : The child process has pid = 22867
[ 7954.765339] [program2] : This is the parent process, pid = 22866
[ 7954.765404] [program2] : child process
[ 7954.828692] [program2] : get SIGINT signal
[ 7954.845349] [program2] : child process terminated
[ 7954.863740] [program2] : The return signal is 2
```

Kill



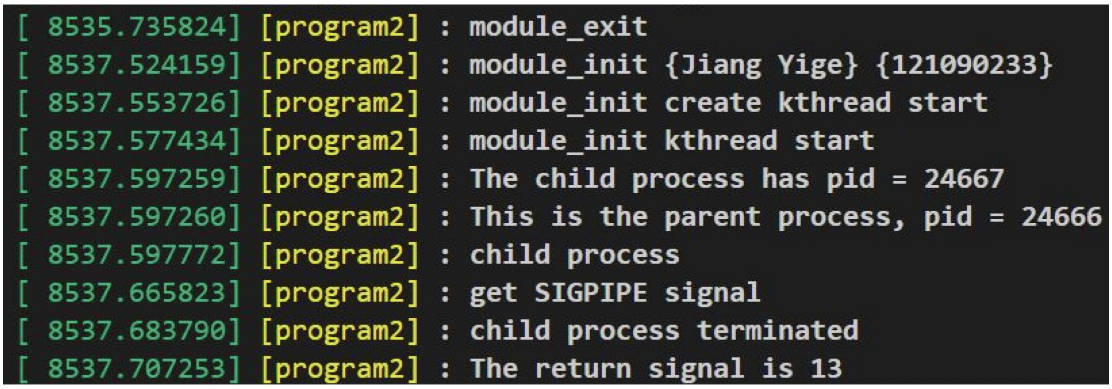
```
[ 8339.658206] [program2] : module_exit
[ 8341.217770] [program2] : module_init {Jiang Yige} {121090233}
[ 8341.239318] [program2] : module_init create kthread start
[ 8341.261942] [program2] : module_init kthread start
[ 8341.282159] [program2] : The child process has pid = 23407
[ 8341.282160] [program2] : This is the parent process, pid = 23406
[ 8341.302123] [program2] : child process
[ 8341.337983] [program2] : get SIGKILL signal
[ 8341.354232] [program2] : child process terminated
[ 8341.375505] [program2] : The return signal is 9
```

Normal



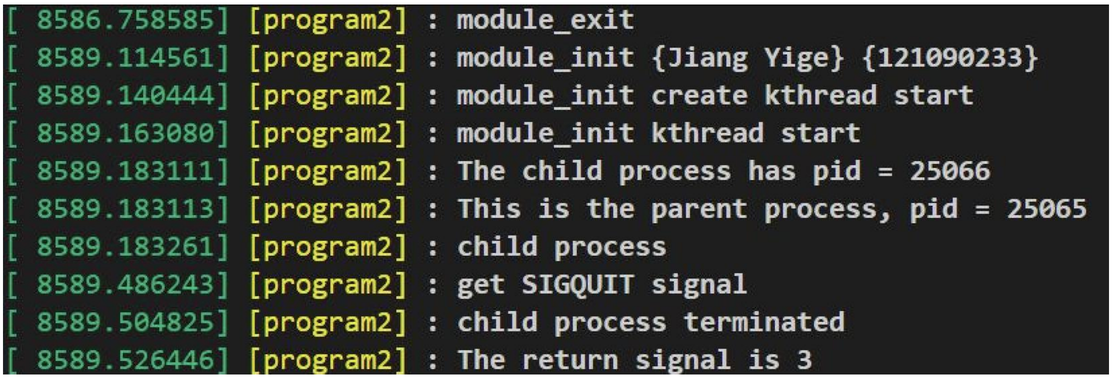
```
[ 8383.782781] [program2] : module_exit
[ 8386.055372] [program2] : module_init {Jiang Yige} {121090233}
[ 8386.078528] [program2] : module_init create kthread start
[ 8386.100891] [program2] : module_init kthread start
[ 8386.118372] [program2] : The child process has pid = 23798
[ 8386.118373] [program2] : This is the parent process, pid = 23797
[ 8386.118675] [program2] : child process
[ 8386.175868] [program2] : Normal termination with EXIT STATUS = 0
[ 8386.200384] [program2] : child process terminated
[ 8386.219175] [program2] : The return signal is 0
```

Pipe



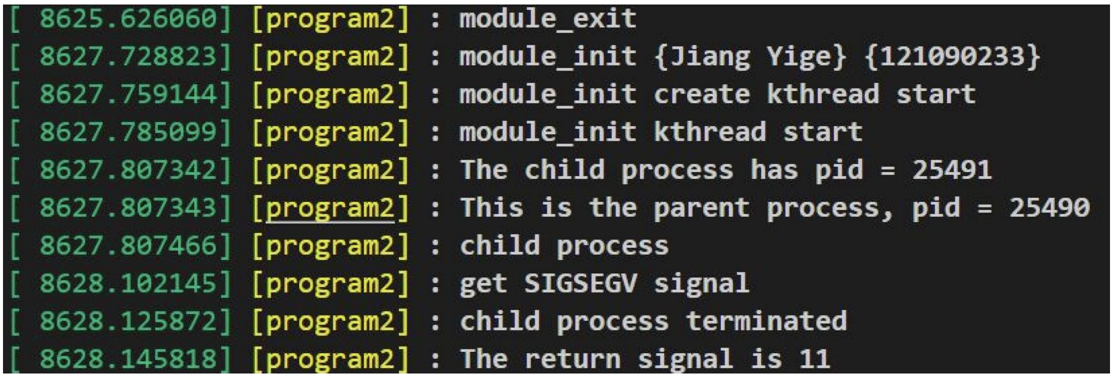
```
[ 8535.735824] [program2] : module_exit
[ 8537.524159] [program2] : module_init {Jiang Yige} {121090233}
[ 8537.553726] [program2] : module_init create kthread start
[ 8537.577434] [program2] : module_init kthread start
[ 8537.597259] [program2] : The child process has pid = 24667
[ 8537.597260] [program2] : This is the parent process, pid = 24666
[ 8537.597772] [program2] : child process
[ 8537.665823] [program2] : get SIGPIPE signal
[ 8537.683790] [program2] : child process terminated
[ 8537.707253] [program2] : The return signal is 13
```

Quit



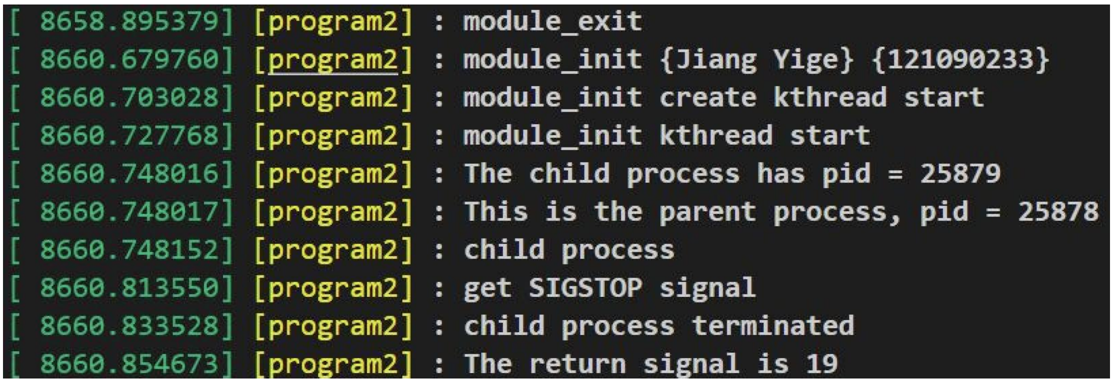
```
[ 8586.758585] [program2] : module_exit
[ 8589.114561] [program2] : module_init {Jiang Yige} {121090233}
[ 8589.140444] [program2] : module_init create kthread start
[ 8589.163080] [program2] : module_init kthread start
[ 8589.183111] [program2] : The child process has pid = 25066
[ 8589.183113] [program2] : This is the parent process, pid = 25065
[ 8589.183261] [program2] : child process
[ 8589.486243] [program2] : get SIGQUIT signal
[ 8589.504825] [program2] : child process terminated
[ 8589.526446] [program2] : The return signal is 3
```

Segment_fault



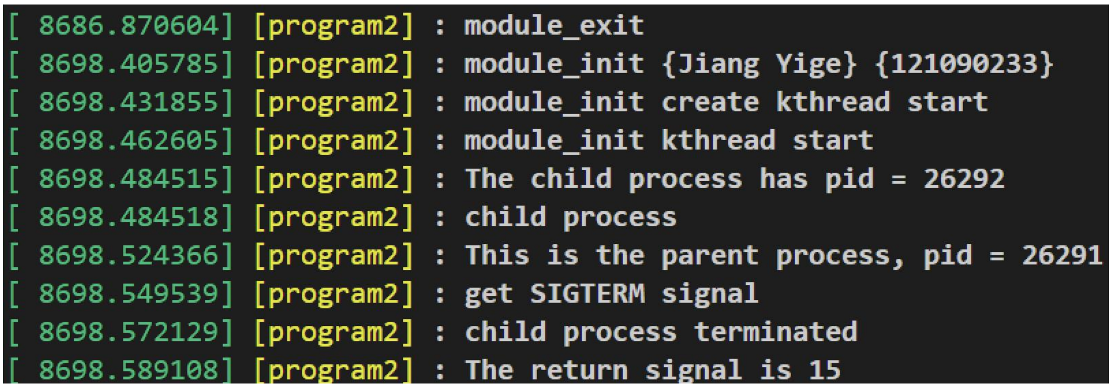
```
[ 8625.626060] [program2] : module_exit
[ 8627.728823] [program2] : module_init {Jiang Yige} {121090233}
[ 8627.759144] [program2] : module_init create kthread start
[ 8627.785099] [program2] : module_init kthread start
[ 8627.807342] [program2] : The child process has pid = 25491
[ 8627.807343] [program2] : This is the parent process, pid = 25490
[ 8627.807466] [program2] : child process
[ 8628.102145] [program2] : get SIGSEGV signal
[ 8628.125872] [program2] : child process terminated
[ 8628.145818] [program2] : The return signal is 11
```

Stop



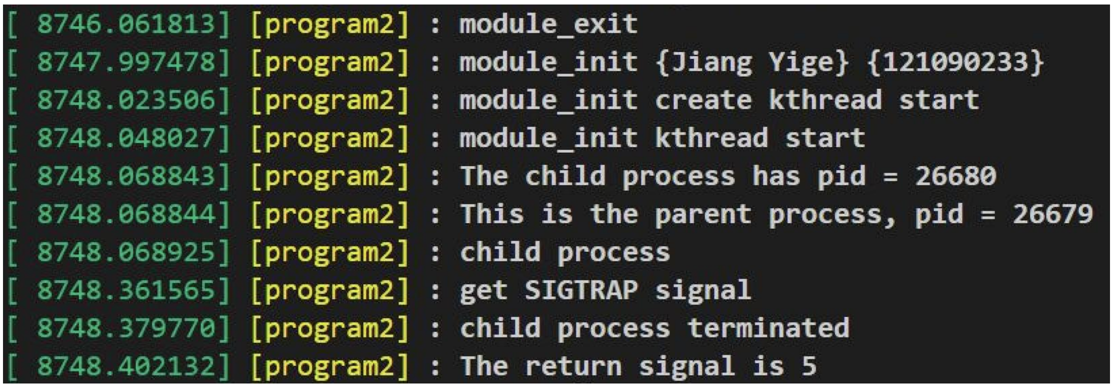
```
[ 8658.895379] [program2] : module_exit
[ 8660.679760] [program2] : module_init {Jiang Yige} {121090233}
[ 8660.703028] [program2] : module_init create kthread start
[ 8660.727768] [program2] : module_init kthread start
[ 8660.748016] [program2] : The child process has pid = 25879
[ 8660.748017] [program2] : This is the parent process, pid = 25878
[ 8660.748152] [program2] : child process
[ 8660.813550] [program2] : get SIGSTOP signal
[ 8660.833528] [program2] : child process terminated
[ 8660.854673] [program2] : The return signal is 19
```

Terminate



```
[ 8686.870604] [program2] : module_exit
[ 8698.405785] [program2] : module_init {Jiang Yige} {121090233}
[ 8698.431855] [program2] : module_init create kthread start
[ 8698.462605] [program2] : module_init kthread start
[ 8698.484515] [program2] : The child process has pid = 26292
[ 8698.484518] [program2] : child process
[ 8698.524366] [program2] : This is the parent process, pid = 26291
[ 8698.549539] [program2] : get SIGTERM signal
[ 8698.572129] [program2] : child process terminated
[ 8698.589108] [program2] : The return signal is 15
```

Trap



```
[ 8746.061813] [program2] : module_exit
[ 8747.997478] [program2] : module_init {Jiang Yige} {121090233}
[ 8748.023506] [program2] : module_init create kthread start
[ 8748.048027] [program2] : module_init kthread start
[ 8748.068843] [program2] : The child process has pid = 26680
[ 8748.068844] [program2] : This is the parent process, pid = 26679
[ 8748.068925] [program2] : child process
[ 8748.361565] [program2] : get SIGTRAP signal
[ 8748.379770] [program2] : child process terminated
[ 8748.402132] [program2] : The return signal is 5
```

What I learned

1. How to change the kernel version and how to compile the kernel.
2. How to implement some functions of *folk(), wait(),* and so on.
3. How to modify kernel sources.
4. Deeper understanding of the process and thread.