Report:CSC3150_Assignment_1

Name:Zhang fengyu

ID:120090825

1. ***Program 1***

(1) Program 1 ideas:

Program1 is about to run the program1.c in the user mode.It will fork a child process,the child process will execute the test program.When it finish,the parent process can receive the signal and then,it can print out related information.It shows the basic idea of the parent and child process.

(2) Design and implementation:

The program1.c contains the main function.Within the main function,firstly it will use fork() function to fork a child process.After that,the program can print out some related information such as pid.Then,it will check the pid and do it by cases.If the pid is -1,it means there are some error.If the pid equals to 0,it shows that is the child process,it will execute the test program and print out related information.In other situations,the parent process can wait for the signal of the child program by using the waitpid() function.Therefore,it can implement the requirement of the execute and wait.Finally,it will print out the status and signal as required.The logic and flow of the whole program are reflected in the different output and judgment conditions.The final result can meet the output requirements.

(3) Environment:

Linux Kernel Version:5.10.146

Linux Distribution:Ubuntu

GCC Version:5.4.0

(4) Setup and execute:

First,go the the folder(/source/program1).

Then,you can type “make” in the terminal.

After that,for different test,you can type “./program1 ./testfile”to test different cases.Here,the testfile should be abort.c,bus.c,normal.c etc.

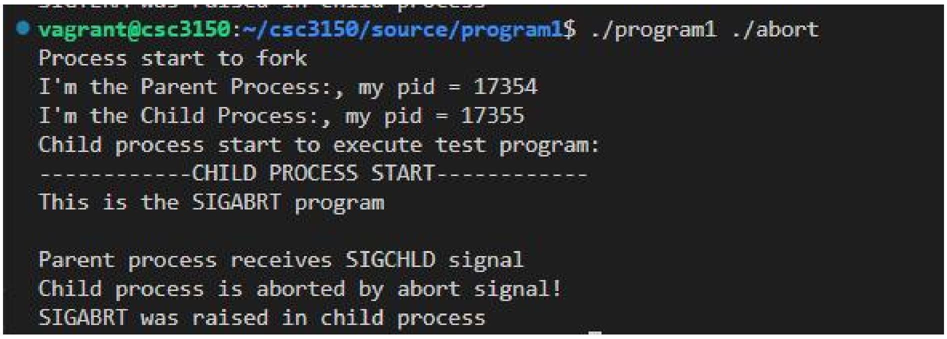
Finally,the terminal will output the information and you can check the result.

(5) What did I learn from Program1:

In general, I learned how to create a child process and get information about the process, such as Pid. In addition, I learned how to use processes to execute different programs and give corresponding signals. In general, I have roughly learned the main process and content of user mode process, and also deepened my understanding of operating system process knowledge.

(6) Output Screenshots:

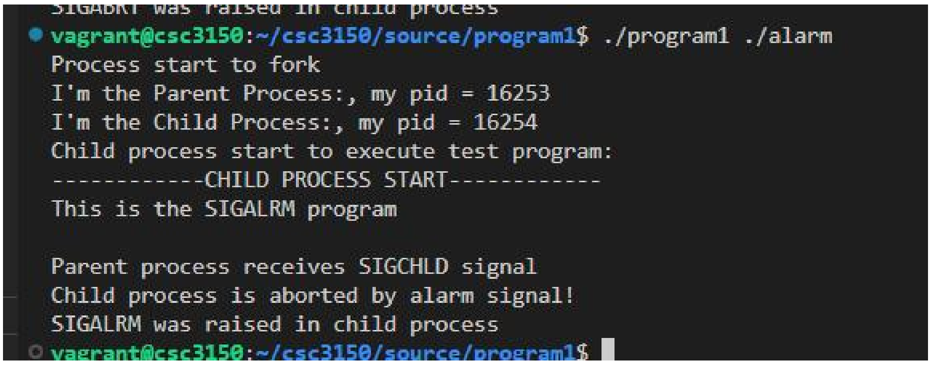
1.abort:



```
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./abort
Process start to fork
I'm the Parent Process:, my pid = 17354
I'm the Child Process:, my pid = 17355
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGABRT program

Parent process receives SIGCHLD signal
Child process is aborted by abort signal!
SIGABRT was raised in child process
```

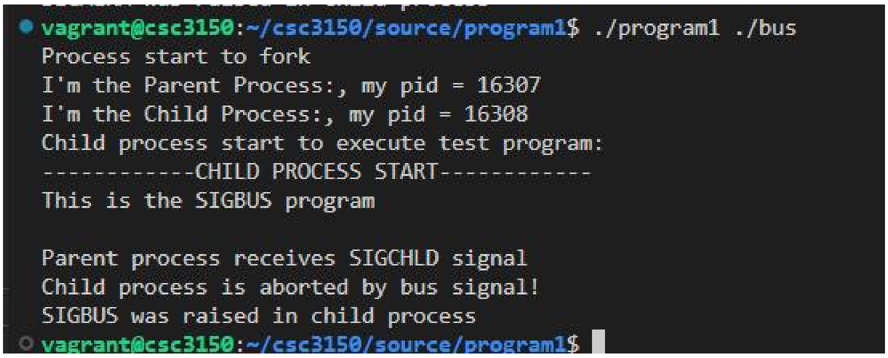
2.alarm:



```
SIGABRT was raised in child process
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./alarm
Process start to fork
I'm the Parent Process:, my pid = 16253
I'm the Child Process:, my pid = 16254
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGALRM program

Parent process receives SIGCHLD signal
Child process is aborted by alarm signal!
SIGALRM was raised in child process
vagrant@csc3150:~/csc3150/source/program1$
```

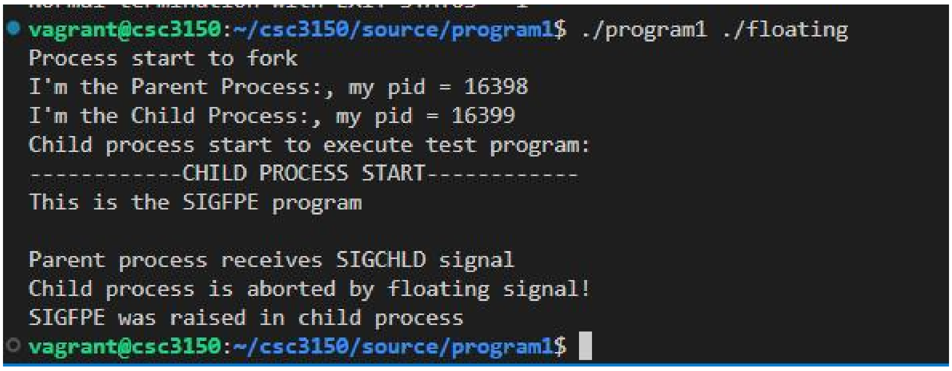
3.bus:



```
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./bus
Process start to fork
I'm the Parent Process:, my pid = 16307
I'm the Child Process:, my pid = 16308
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGBUS program

Parent process receives SIGCHLD signal
Child process is aborted by bus signal!
SIGBUS was raised in child process
vagrant@csc3150:~/csc3150/source/program1$
```

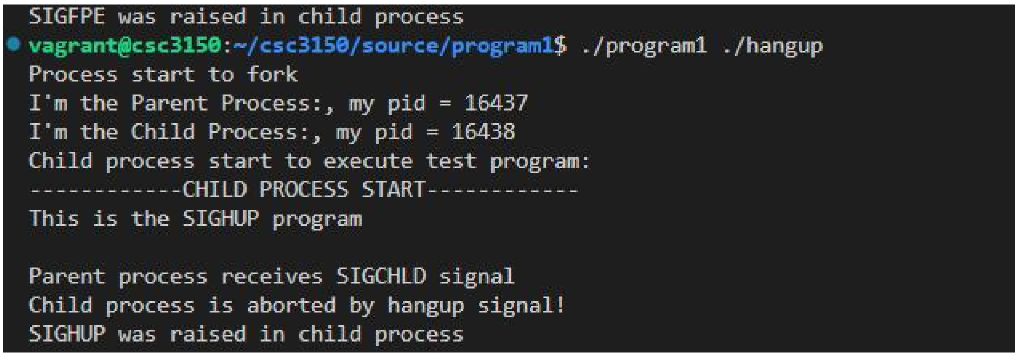
4. Float:



```
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./floating
Process start to fork
I'm the Parent Process:, my pid = 16398
I'm the Child Process:, my pid = 16399
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGFPE program

Parent process receives SIGCHLD signal
Child process is aborted by floating signal!
SIGFPE was raised in child process
vagrant@csc3150:~/csc3150/source/program1$
```

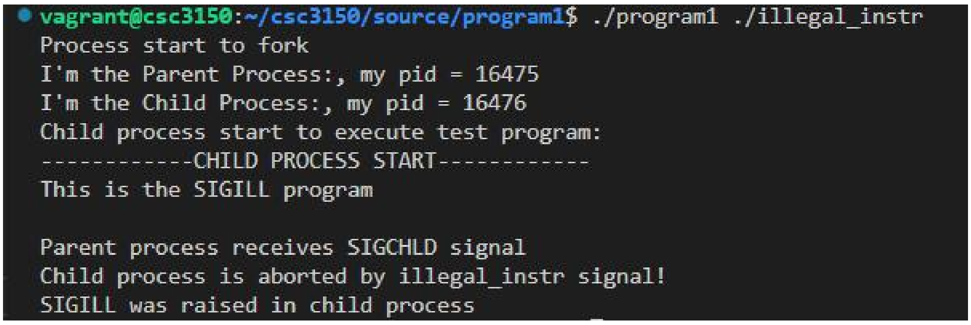
5. Hangup:



```
SIGFPE was raised in child process
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./hangup
Process start to fork
I'm the Parent Process:, my pid = 16437
I'm the Child Process:, my pid = 16438
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGHUP program

Parent process receives SIGCHLD signal
Child process is aborted by hangup signal!
SIGHUP was raised in child process
```

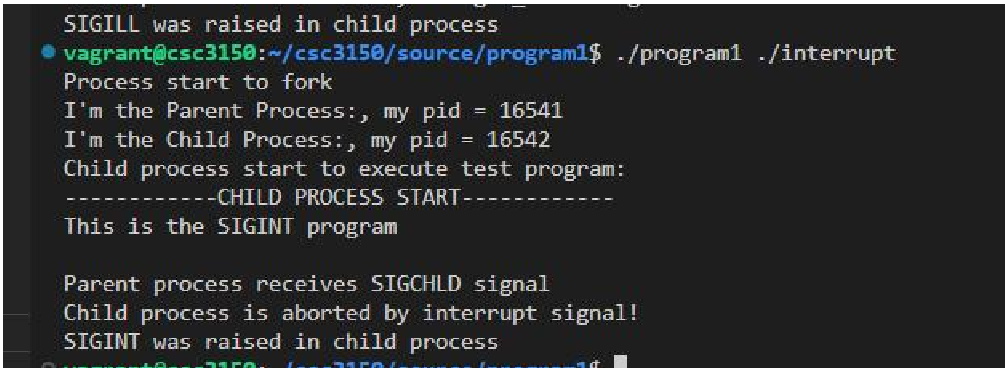
6. illegal_instruction:



```
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./illegal_instr
Process start to fork
I'm the Parent Process:, my pid = 16475
I'm the Child Process:, my pid = 16476
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGILL program

Parent process receives SIGCHLD signal
Child process is aborted by illegal_instr signal!
SIGILL was raised in child process
```

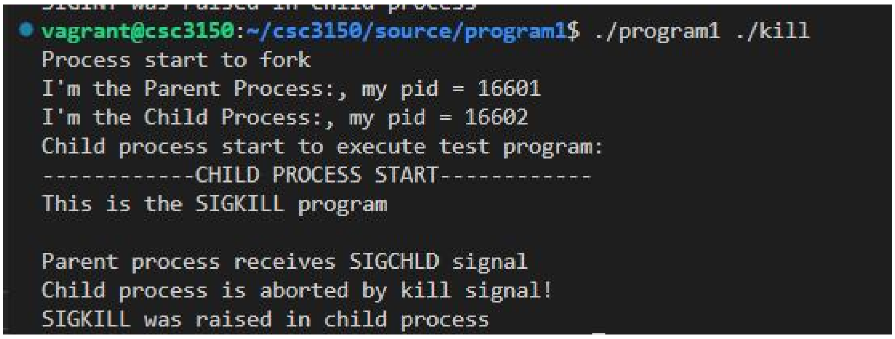
7. interrupt:



```
SIGILL was raised in child process
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./interrupt
Process start to fork
I'm the Parent Process:, my pid = 16541
I'm the Child Process:, my pid = 16542
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGINT program

Parent process receives SIGCHLD signal
Child process is aborted by interrupt signal!
SIGINT was raised in child process
vagrant@csc3150:~/csc3150/source/program1$
```

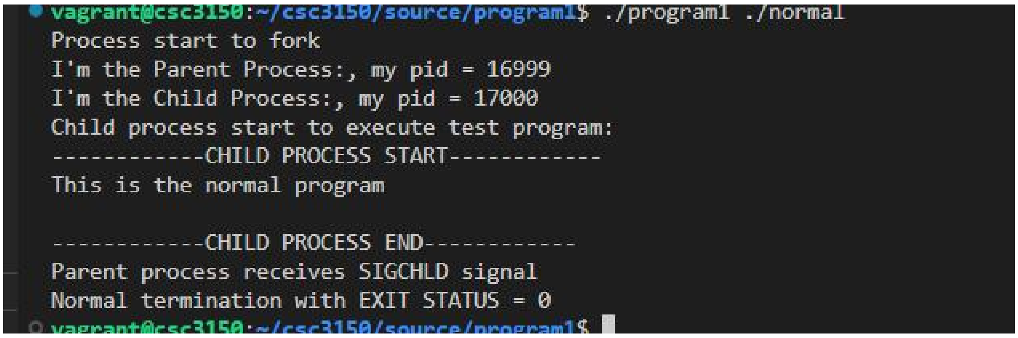
8. kill:



```
SIGINT was raised in child process
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./kill
Process start to fork
I'm the Parent Process:, my pid = 16601
I'm the Child Process:, my pid = 16602
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGKILL program

Parent process receives SIGCHLD signal
Child process is aborted by kill signal!
SIGKILL was raised in child process
```

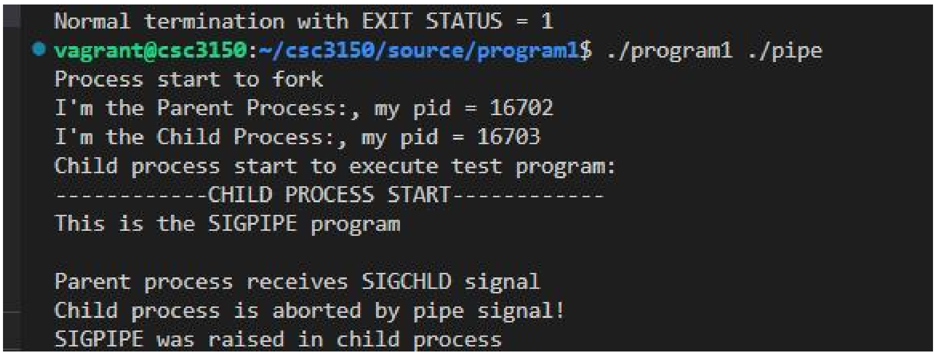
9. normal:



```
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./normal
Process start to fork
I'm the Parent Process:, my pid = 16999
I'm the Child Process:, my pid = 17000
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:~/csc3150/source/program1$
```

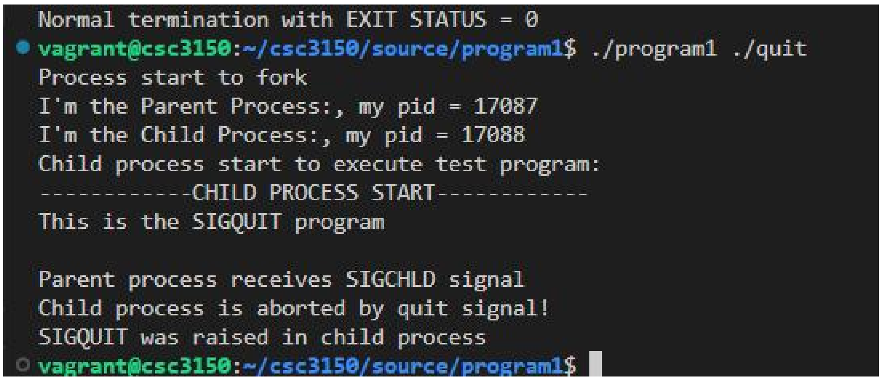
10. pipe:



```
Normal termination with EXIT STATUS = 1
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./pipe
Process start to fork
I'm the Parent Process:, my pid = 16702
I'm the Child Process:, my pid = 16703
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGPIPE program

Parent process receives SIGCHLD signal
Child process is aborted by pipe signal!
SIGPIPE was raised in child process
```

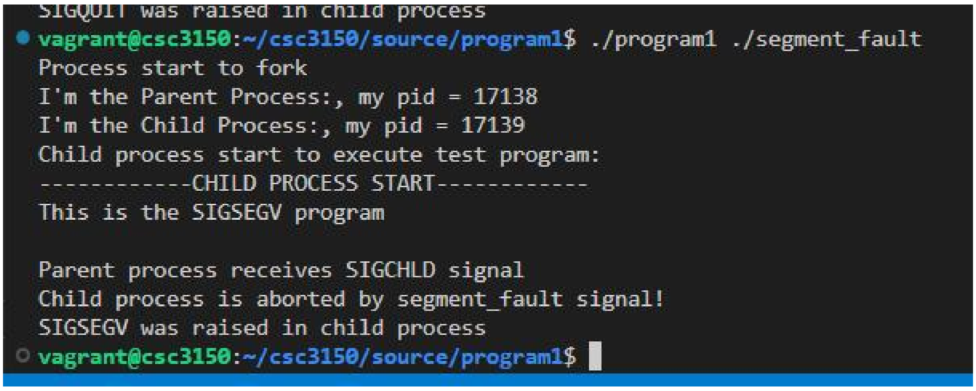
11. quit:



```
Normal termination with EXIT STATUS = 0
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./quit
Process start to fork
I'm the Parent Process:, my pid = 17087
I'm the Child Process:, my pid = 17088
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGQUIT program

Parent process receives SIGCHLD signal
Child process is aborted by quit signal!
SIGQUIT was raised in child process
vagrant@csc3150:~/csc3150/source/program1$
```

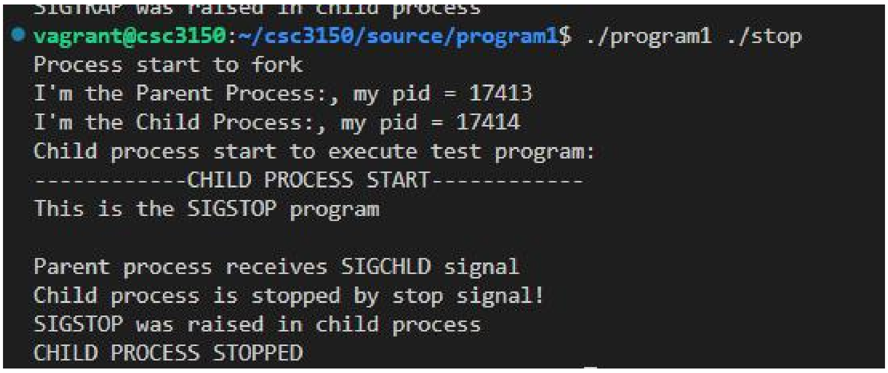
12. segment fault:



```
SIGQUIT was raised in child process
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./segment_fault
Process start to fork
I'm the Parent Process:, my pid = 17138
I'm the Child Process:, my pid = 17139
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGSEGV program

Parent process receives SIGCHLD signal
Child process is aborted by segment_fault signal!
SIGSEGV was raised in child process
vagrant@csc3150:~/csc3150/source/program1$
```

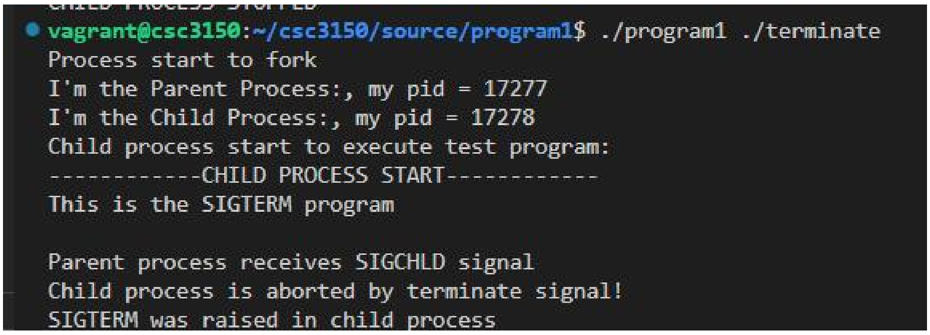
13. stop:



```
SIGTRAP was raised in child process
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./stop
Process start to fork
I'm the Parent Process:, my pid = 17413
I'm the Child Process:, my pid = 17414
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGSTOP program

Parent process receives SIGCHLD signal
Child process is stopped by stop signal!
SIGSTOP was raised in child process
CHILD PROCESS STOPPED
```

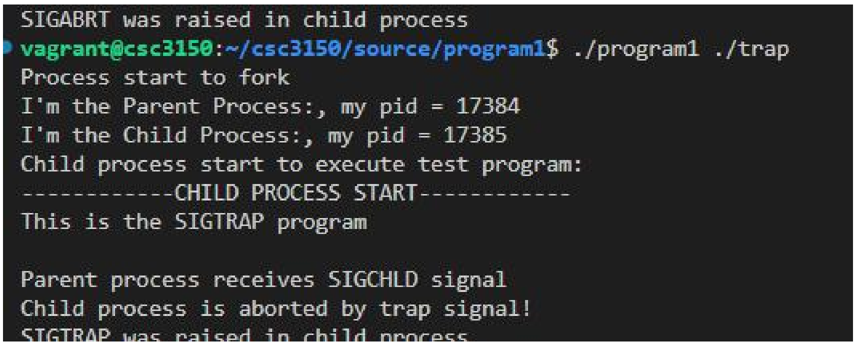
14. terminate:



```
CHILD PROCESS STOPPED
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./terminate
Process start to fork
I'm the Parent Process:, my pid = 17277
I'm the Child Process:, my pid = 17278
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGTERM program

Parent process receives SIGCHLD signal
Child process is aborted by terminate signal!
SIGTERM was raised in child process
```

15. trap:



```
SIGABRT was raised in child process
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./trap
Process start to fork
I'm the Parent Process:, my pid = 17384
I'm the Child Process:, my pid = 17385
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGTRAP program

Parent process receives SIGCHLD signal
Child process is aborted by trap signal!
SIGTRAP was raised in child process
```

2. ***Program 2***

(1) Program 2 ideas:

Program2 is abort to create a kernel thread.Then,it will fork a child process,the child process execute another test program.The same idea as above,the process will wait for the child process,it can get the signal and status and it will print out related information.

(2) Design and implementation:

At the very beginning,we need to change some part of the kernel.We need to use 4 functions:_do_fork(),do_exec(),getname_kernel(),do_wait(),we need to go to the linux file and export then by adding EXPORT_SYMBOL.Then,we need to compile the kernel.Therefore,we can use this function in our program2.c.Also,we need to extern them.This is the first thing to do before dealing with the task.

Then,we need to write the program2.c.This is the program we used for the task,we also have a test.c to test for it.We implement the task using different function.We have my_fork(),this is the main part and logic of our program.Firstly,we use kernel_thread() to create a kernel thread,it run my_fork(),it will fork a child process.

There,we use my_exec() function to execute the test file.It use getname_kernel to do this,the path I used to test is /home/vagrant/csc3150/source/program2/test.As required,I change it to /tmp/test.

We have my_wait() function to deal with different test case and output different information.It use struct wait_opts and do_wait to make it acceptable.Different cases are processed by IF judgment and corresponding results are output.In the program,we also have some basic function for initialization and exit.

Overall,we successfully create the thread and child process,execute the test program and output the correct result.

(3) Environment:

Linux Kernel Version:5.10.146

Linux Distribution:Ubuntu

GCC Version:5.4.0

(4) Setup and execute:

First,for the kernel part.We need to export the 4 functions.Then,we go to the terminal,use “make bzImage”, “make modules” , “make modules_install” , “make install” and “reboot”.

Then,you can go the the program2 folder and use gcc -o test test.c to compile test file.Then,type “sudo make” in the terminal.

After that,for different test,you can type “sudo insmod program2.ko” , “sudo rmmod program2.ko”,and use “dmesg -c” to get the display.Sometimes for the first time it may display lots of status information,you can just use “make clean” and retry the above command,then you can get the correct output.

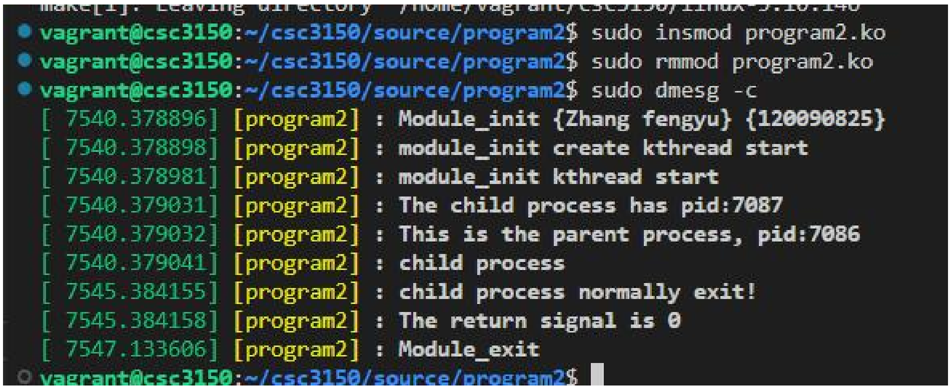
Finally,the terminal will output the information and you can check the result.

(5) What did I learn from Program2:

In general, I learned how to look at the kernel source code and make minor changes. In addition, I know how to create processes and process files in kernel mode, and output the corresponding state and content.In addition, I learned how to write simple and explicit programs to accomplish tasks in this environment. This reinforces my understanding of the operating system kernel.

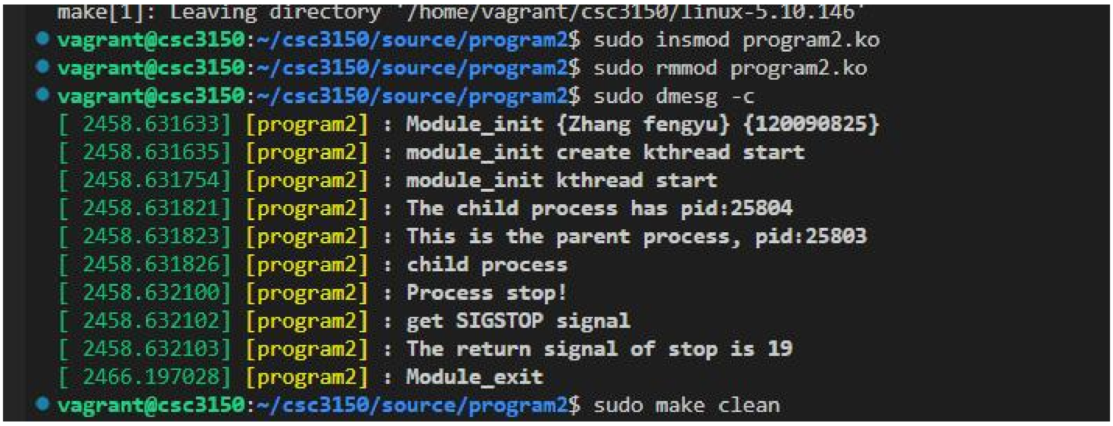
(6) Output Screenshots:

1. Normal:



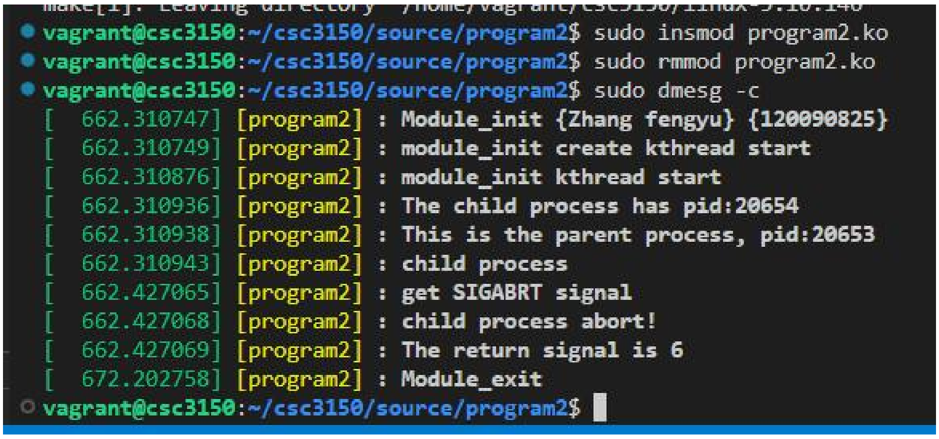
```
  make[1]: Leaving directory '/home/vagrant/csc3150/linux-5.10.146
● vagrant@csc3150:~/csc3150/source/program2$ sudo insmod program2.ko
● vagrant@csc3150:~/csc3150/source/program2$ sudo rmmod program2.ko
● vagrant@csc3150:~/csc3150/source/program2$ sudo dmesg -c
  [ 7540.378896] [program2] : Module_init {Zhang fengyu} {120090825}
  [ 7540.378898] [program2] : module_init create kthread start
  [ 7540.378981] [program2] : module_init kthread start
  [ 7540.379031] [program2] : The child process has pid:7087
  [ 7540.379032] [program2] : This is the parent process, pid:7086
  [ 7540.379041] [program2] : child process
  [ 7545.384155] [program2] : child process normally exit!
  [ 7545.384158] [program2] : The return signal is 0
  [ 7547.133606] [program2] : Module_exit
○ vagrant@csc3150:~/csc3150/source/program2$ 
```

2. Stop:



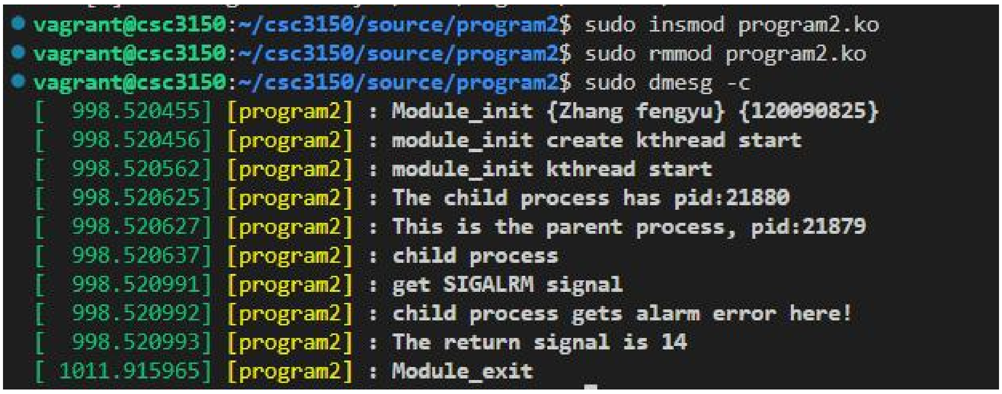
```
  make[1]: Leaving directory '/home/vagrant/csc3150/linux-5.10.146'
vagrant@csc3150:~/csc3150/source/program2$ sudo insmod program2.ko
vagrant@csc3150:~/csc3150/source/program2$ sudo rmmod program2.ko
vagrant@csc3150:~/csc3150/source/program2$ sudo dmesg -c
[ 2458.631633] [program2] : Module_init {Zhang fengyu} {120090825}
[ 2458.631635] [program2] : module_init create kthread start
[ 2458.631754] [program2] : module_init kthread start
[ 2458.631821] [program2] : The child process has pid:25804
[ 2458.631823] [program2] : This is the parent process, pid:25803
[ 2458.631826] [program2] : child process
[ 2458.632100] [program2] : Process stop!
[ 2458.632102] [program2] : get SIGSTOP signal
[ 2458.632103] [program2] : The return signal of stop is 19
[ 2466.197028] [program2] : Module_exit
vagrant@csc3150:~/csc3150/source/program2$ sudo make clean
```

3. abort:



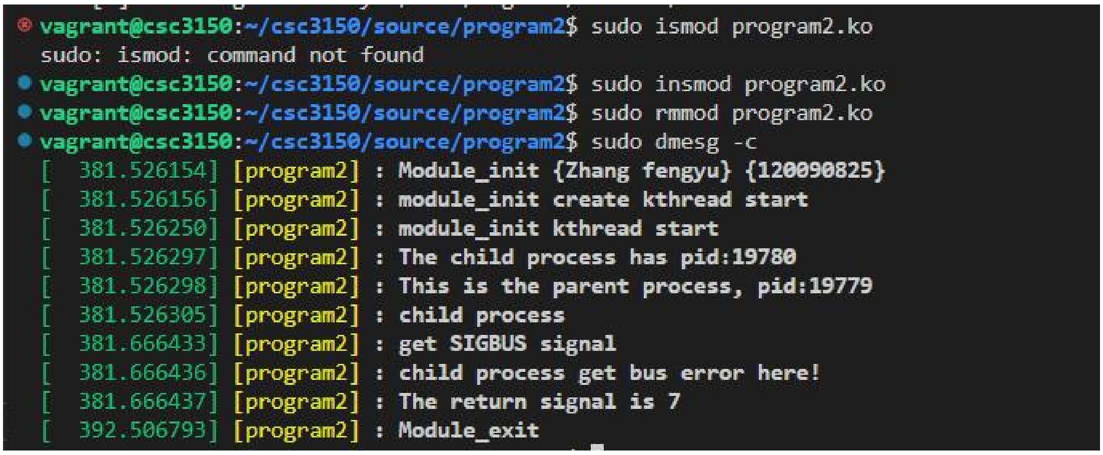
```
make[1]: Leaving directory '/home/vagrant/csc3150/linux-5.10.146'
vagrant@csc3150:~/csc3150/source/program2$ sudo insmod program2.ko
vagrant@csc3150:~/csc3150/source/program2$ sudo rmmod program2.ko
vagrant@csc3150:~/csc3150/source/program2$ sudo dmesg -c
[  662.310747] [program2] : Module_init {Zhang fengyu} {120090825}
[  662.310749] [program2] : module_init create kthread start
[  662.310876] [program2] : module_init kthread start
[  662.310936] [program2] : The child process has pid:20654
[  662.310938] [program2] : This is the parent process, pid:20653
[  662.310943] [program2] : child process
[  662.427065] [program2] : get SIGABRT signal
[  662.427068] [program2] : child process abort!
[  662.427069] [program2] : The return signal is 6
[  672.202758] [program2] : Module_exit
vagrant@csc3150:~/csc3150/source/program2$
```

4. alarm:



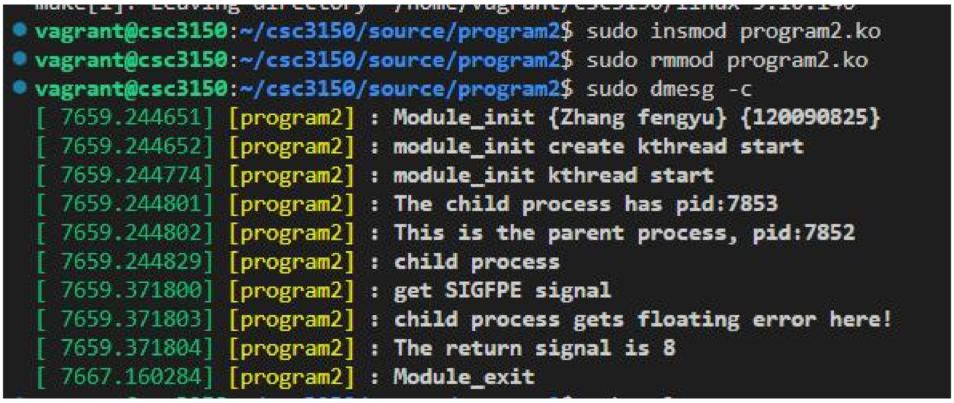
```
vagrant@csc3150:~/csc3150/source/program2$ sudo insmod program2.ko
vagrant@csc3150:~/csc3150/source/program2$ sudo rmmod program2.ko
vagrant@csc3150:~/csc3150/source/program2$ sudo dmesg -c
[  998.520455] [program2] : Module_init {Zhang fengyu} {120090825}
[  998.520456] [program2] : module_init create kthread start
[  998.520562] [program2] : module_init kthread start
[  998.520625] [program2] : The child process has pid:21880
[  998.520627] [program2] : This is the parent process, pid:21879
[  998.520637] [program2] : child process
[  998.520991] [program2] : get SIGALRM signal
[  998.520992] [program2] : child process gets alarm error here!
[  998.520993] [program2] : The return signal is 14
[ 1011.915965] [program2] : Module_exit
```

5. bus:



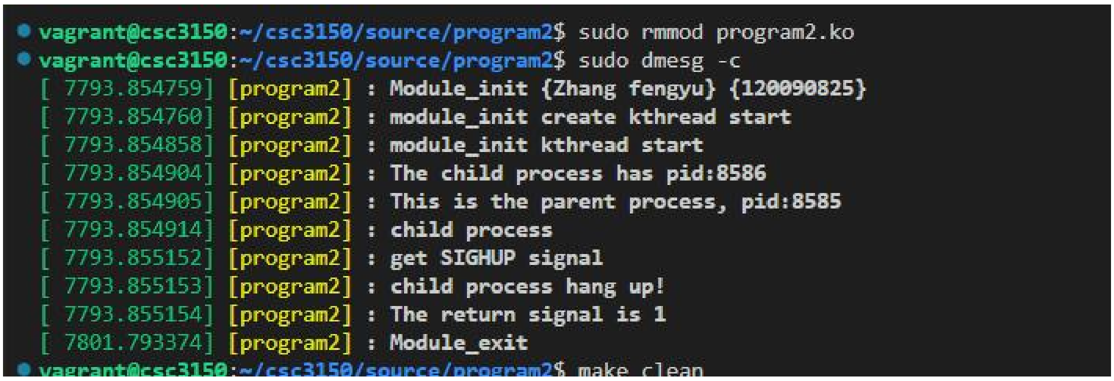
```
vagrant@csc3150:~/csc3150/source/program2$ sudo ismod program2.ko
sudo: ismod: command not found
vagrant@csc3150:~/csc3150/source/program2$ sudo insmod program2.ko
vagrant@csc3150:~/csc3150/source/program2$ sudo rmmod program2.ko
vagrant@csc3150:~/csc3150/source/program2$ sudo dmesg -c
[  381.526154] [program2] : Module_init {Zhang fengyu} {120090825}
[  381.526156] [program2] : module_init create kthread start
[  381.526250] [program2] : module_init kthread start
[  381.526297] [program2] : The child process has pid:19780
[  381.526298] [program2] : This is the parent process, pid:19779
[  381.526305] [program2] : child process
[  381.666433] [program2] : get SIGBUS signal
[  381.666436] [program2] : child process get bus error here!
[  381.666437] [program2] : The return signal is 7
[  392.506793] [program2] : Module_exit
```

6. floating:



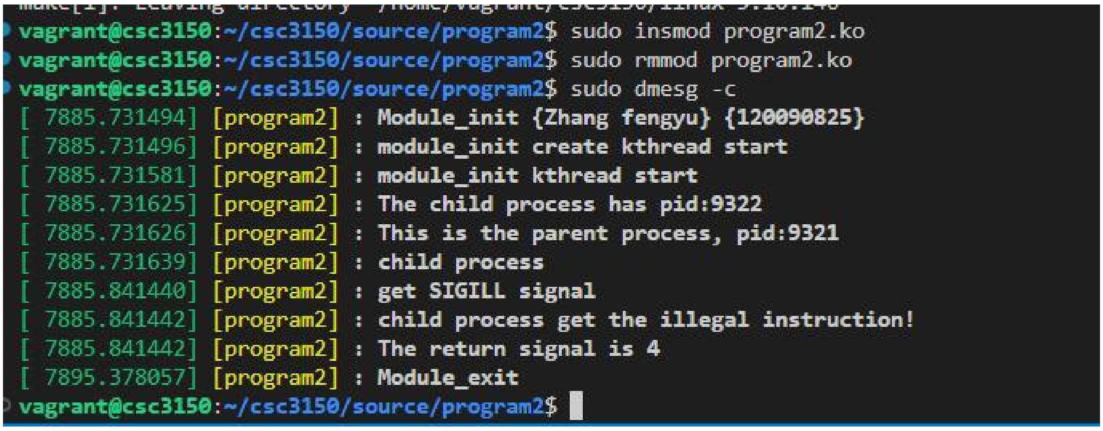
```
make[1]: Leaving directory '/home/vagrant/csc3150/linux-5.10.146'
vagrant@csc3150:~/csc3150/source/program2$ sudo insmod program2.ko
vagrant@csc3150:~/csc3150/source/program2$ sudo rmmod program2.ko
vagrant@csc3150:~/csc3150/source/program2$ sudo dmesg -c
[ 7659.244651] [program2] : Module_init {Zhang fengyu} {120090825}
[ 7659.244652] [program2] : module_init create kthread start
[ 7659.244774] [program2] : module_init kthread start
[ 7659.244801] [program2] : The child process has pid:7853
[ 7659.244802] [program2] : This is the parent process, pid:7852
[ 7659.244829] [program2] : child process
[ 7659.371800] [program2] : get SIGFPE signal
[ 7659.371803] [program2] : child process gets floating error here!
[ 7659.371804] [program2] : The return signal is 8
[ 7667.160284] [program2] : Module_exit
```

7. hangup:



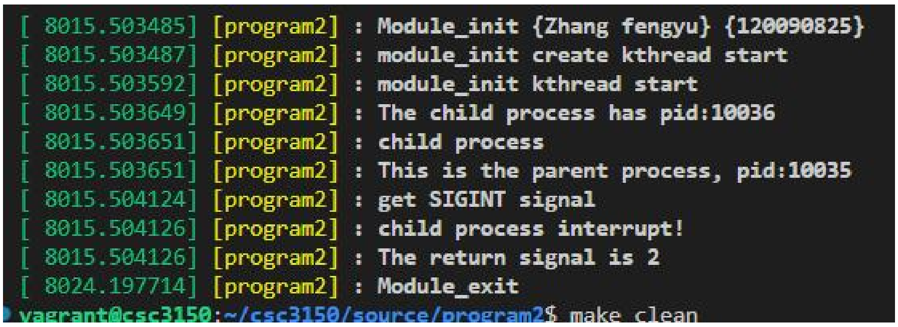
```
vagrant@csc3150:~/csc3150/source/program2$ sudo rmmod program2.ko
vagrant@csc3150:~/csc3150/source/program2$ sudo dmesg -c
[ 7793.854759] [program2] : Module_init {Zhang fengyu} {120090825}
[ 7793.854760] [program2] : module_init create kthread start
[ 7793.854858] [program2] : module_init kthread start
[ 7793.854904] [program2] : The child process has pid:8586
[ 7793.854905] [program2] : This is the parent process, pid:8585
[ 7793.854914] [program2] : child process
[ 7793.855152] [program2] : get SIGHUP signal
[ 7793.855153] [program2] : child process hang up!
[ 7793.855154] [program2] : The return signal is 1
[ 7801.793374] [program2] : Module_exit
vagrant@csc3150:~/csc3150/source/program2$ make clean
```

8. illegal instruction:



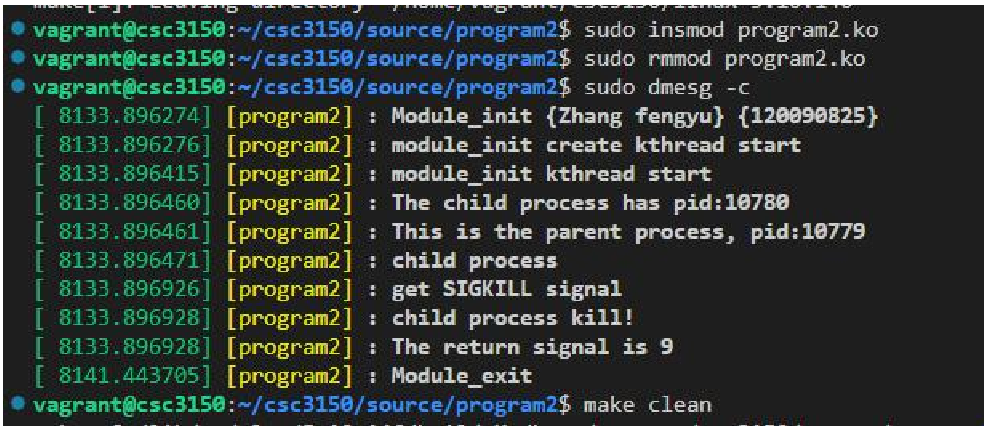
```
vagrant@csc3150:~/csc3150/source/program2$ sudo insmod program2.ko
vagrant@csc3150:~/csc3150/source/program2$ sudo rmmod program2.ko
vagrant@csc3150:~/csc3150/source/program2$ sudo dmesg -c
[ 7885.731494] [program2] : Module_init {Zhang fengyu} {120090825}
[ 7885.731496] [program2] : module_init create kthread start
[ 7885.731581] [program2] : module_init kthread start
[ 7885.731625] [program2] : The child process has pid:9322
[ 7885.731626] [program2] : This is the parent process, pid:9321
[ 7885.731639] [program2] : child process
[ 7885.841440] [program2] : get SIGILL signal
[ 7885.841442] [program2] : child process get the illegal instruction!
[ 7885.841442] [program2] : The return signal is 4
[ 7895.378057] [program2] : Module_exit
vagrant@csc3150:~/csc3150/source/program2$ 
```

9. interrupt:



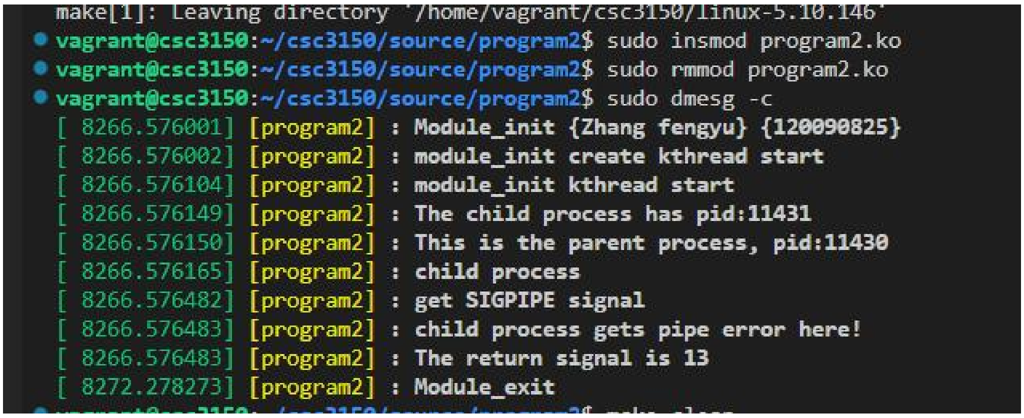
```
[ 8015.503485] [program2] : Module_init {Zhang fengyu} {120090825}
[ 8015.503487] [program2] : module_init create kthread start
[ 8015.503592] [program2] : module_init kthread start
[ 8015.503649] [program2] : The child process has pid:10036
[ 8015.503651] [program2] : child process
[ 8015.503651] [program2] : This is the parent process, pid:10035
[ 8015.504124] [program2] : get SIGINT signal
[ 8015.504126] [program2] : child process interrupt!
[ 8015.504126] [program2] : The return signal is 2
[ 8024.197714] [program2] : Module_exit
vagrant@csc3150:~/csc3150/source/program2$ make clean
```

10. kill:



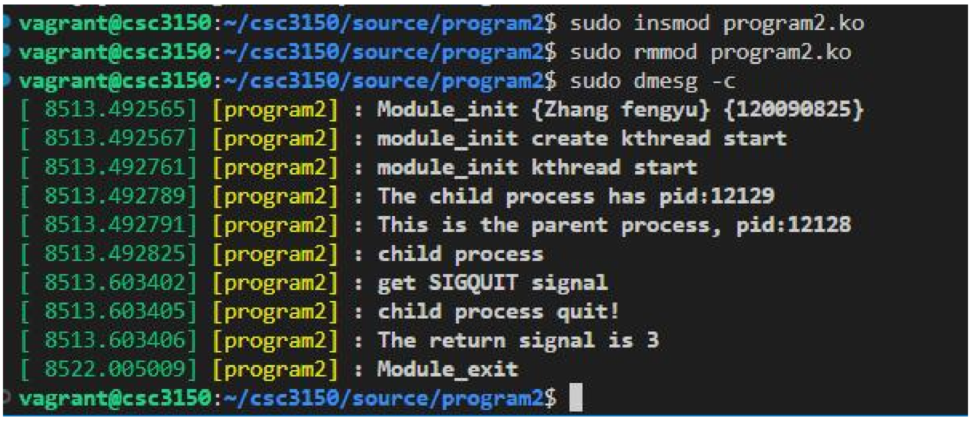
```
vagrant@csc3150:~/csc3150/source/program2$ sudo insmod program2.ko
vagrant@csc3150:~/csc3150/source/program2$ sudo rmmod program2.ko
vagrant@csc3150:~/csc3150/source/program2$ sudo dmesg -c
 [ 8133.896274] [program2] : Module_init {Zhang fengyu} {120090825}
 [ 8133.896276] [program2] : module_init create kthread start
 [ 8133.896415] [program2] : module_init kthread start
 [ 8133.896460] [program2] : The child process has pid:10780
 [ 8133.896461] [program2] : This is the parent process, pid:10779
 [ 8133.896471] [program2] : child process
 [ 8133.896926] [program2] : get SIGKILL signal
 [ 8133.896928] [program2] : child process kill!
 [ 8133.896928] [program2] : The return signal is 9
 [ 8141.443705] [program2] : Module_exit
vagrant@csc3150:~/csc3150/source/program2$ make clean
```

11. pipe:



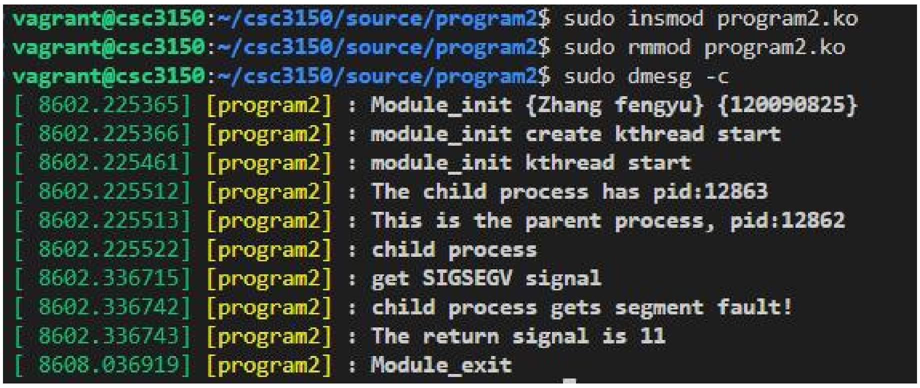
```
make[1]: Leaving directory '/home/vagrant/csc3150/linux-5.10.146'
vagrant@csc3150:~/csc3150/source/program2$ sudo insmod program2.ko
vagrant@csc3150:~/csc3150/source/program2$ sudo rmmod program2.ko
vagrant@csc3150:~/csc3150/source/program2$ sudo dmesg -c
[ 8266.576001] [program2] : Module_init {Zhang fengyu} {120090825}
[ 8266.576002] [program2] : module_init create kthread start
[ 8266.576104] [program2] : module_init kthread start
[ 8266.576149] [program2] : The child process has pid:11431
[ 8266.576150] [program2] : This is the parent process, pid:11430
[ 8266.576165] [program2] : child process
[ 8266.576482] [program2] : get SIGPIPE signal
[ 8266.576483] [program2] : child process gets pipe error here!
[ 8266.576483] [program2] : The return signal is 13
[ 8272.278273] [program2] : Module_exit
```

12. quit:



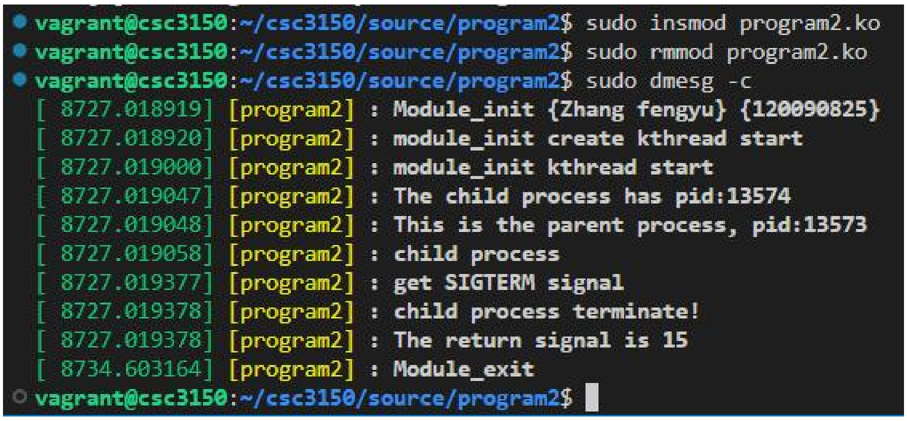
```
vagrant@csc3150:~/csc3150/source/program2$ sudo insmod program2.ko
vagrant@csc3150:~/csc3150/source/program2$ sudo rmmod program2.ko
vagrant@csc3150:~/csc3150/source/program2$ sudo dmesg -c
[ 8513.492565] [program2] : Module_init {Zhang fengyu} {120090825}
[ 8513.492567] [program2] : module_init create kthread start
[ 8513.492761] [program2] : module_init kthread start
[ 8513.492789] [program2] : The child process has pid:12129
[ 8513.492791] [program2] : This is the parent process, pid:12128
[ 8513.492825] [program2] : child process
[ 8513.603402] [program2] : get SIGQUIT signal
[ 8513.603405] [program2] : child process quit!
[ 8513.603406] [program2] : The return signal is 3
[ 8522.005009] [program2] : Module_exit
vagrant@csc3150:~/csc3150/source/program2$
```

13. segment fault:



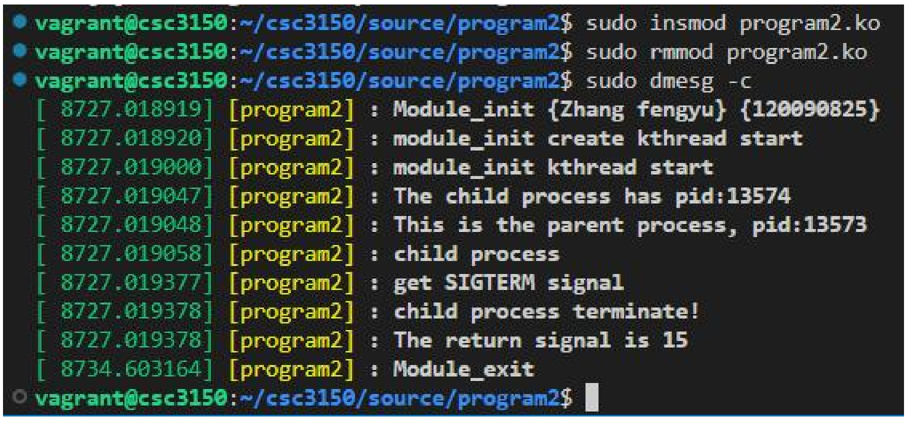
```
vagrant@csc3150:~/csc3150/source/program2$ sudo insmod program2.ko
vagrant@csc3150:~/csc3150/source/program2$ sudo rmmod program2.ko
vagrant@csc3150:~/csc3150/source/program2$ sudo dmesg -c
[ 8602.225365] [program2] : Module_init {Zhang fengyu} {120090825}
[ 8602.225366] [program2] : module_init create kthread start
[ 8602.225461] [program2] : module_init kthread start
[ 8602.225512] [program2] : The child process has pid:12863
[ 8602.225513] [program2] : This is the parent process, pid:12862
[ 8602.225522] [program2] : child process
[ 8602.336715] [program2] : get SIGSEGV signal
[ 8602.336742] [program2] : child process gets segment fault!
[ 8602.336743] [program2] : The return signal is 11
[ 8608.036919] [program2] : Module_exit
```

14. terminate:



```
vagrant@csc3150:~/csc3150/source/program2$ sudo insmod program2.ko
vagrant@csc3150:~/csc3150/source/program2$ sudo rmmod program2.ko
vagrant@csc3150:~/csc3150/source/program2$ sudo dmesg -c
[ 8727.018919] [program2] : Module_init {Zhang fengyu} {120090825}
[ 8727.018920] [program2] : module_init create kthread start
[ 8727.019000] [program2] : module_init kthread start
[ 8727.019047] [program2] : The child process has pid:13574
[ 8727.019048] [program2] : This is the parent process, pid:13573
[ 8727.019058] [program2] : child process
[ 8727.019377] [program2] : get SIGTERM signal
[ 8727.019378] [program2] : child process terminate!
[ 8727.019378] [program2] : The return signal is 15
[ 8734.603164] [program2] : Module_exit
vagrant@csc3150:~/csc3150/source/program2$ 
```

15. trap:



```
vagrant@csc3150:~/csc3150/source/program2$ sudo insmod program2.ko
vagrant@csc3150:~/csc3150/source/program2$ sudo rmmod program2.ko
vagrant@csc3150:~/csc3150/source/program2$ sudo dmesg -c
[ 8727.018919] [program2] : Module_init {Zhang fengyu} {120090825}
[ 8727.018920] [program2] : module_init create kthread start
[ 8727.019000] [program2] : module_init kthread start
[ 8727.019047] [program2] : The child process has pid:13574
[ 8727.019048] [program2] : This is the parent process, pid:13573
[ 8727.019058] [program2] : child process
[ 8727.019377] [program2] : get SIGTERM signal
[ 8727.019378] [program2] : child process terminate!
[ 8727.019378] [program2] : The return signal is 15
[ 8734.603164] [program2] : Module_exit
vagrant@csc3150:~/csc3150/source/program2$ 
```

Report END