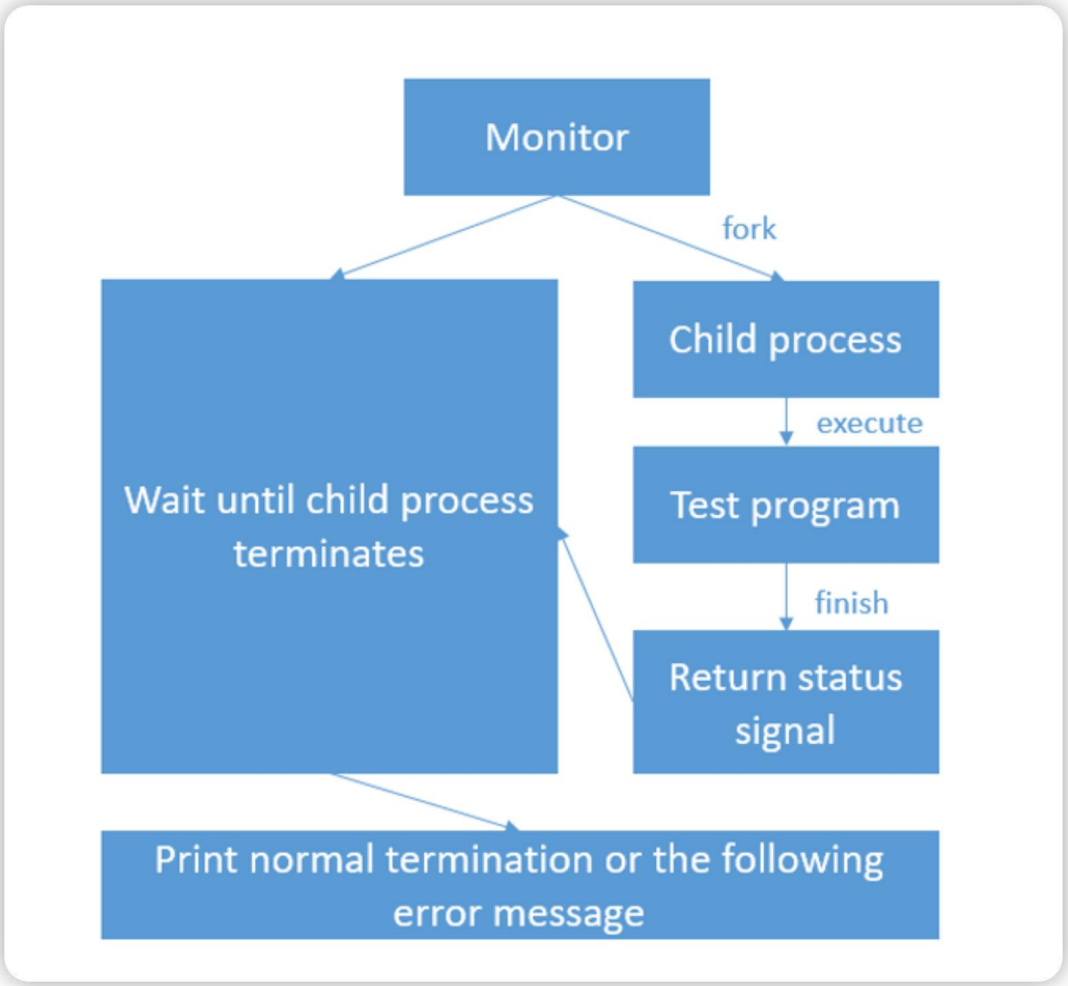
Wang Shijie 120090331

Task1

Program idea:

The most important part of task in task1 is to use user mode to fork a child process to execute the test program. Therefore, the program should create the child process by fork() function. I use the pid_t type to record the process identifier so that the parent process can use wait() function to receives signals from the child process. By the use of execve() function child process can run the test program. Finally, using the WIFEXITED and other similar function can detect the signal and analyze it and give the response by if branch.

The basic process is shown as below:



Environment:

We need a basic linux environment.

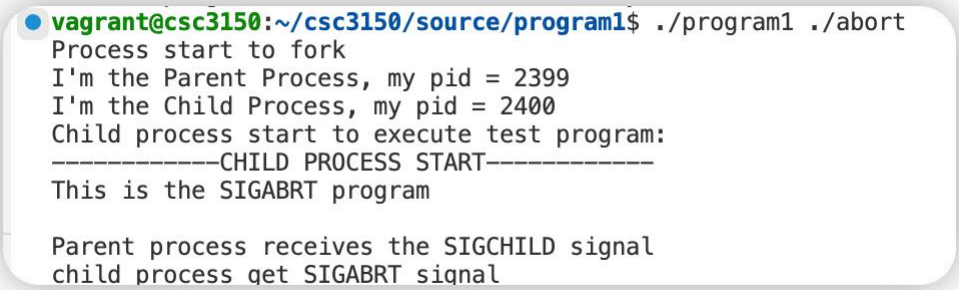
Linux Distribution: Ubuntu 20.04

Linux Kernel Version: 5.10.5

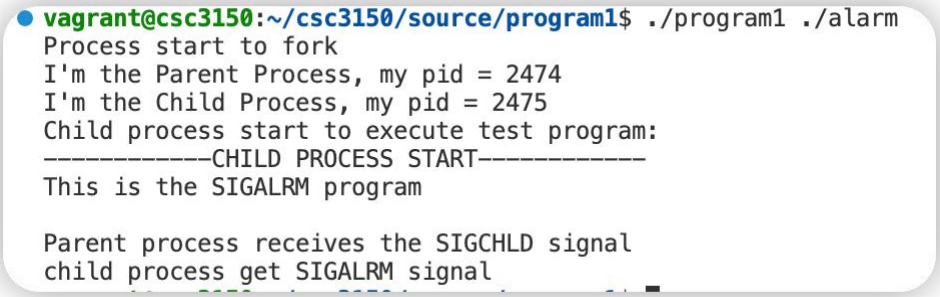
GCC Version: 5.4.0

Screenshot:

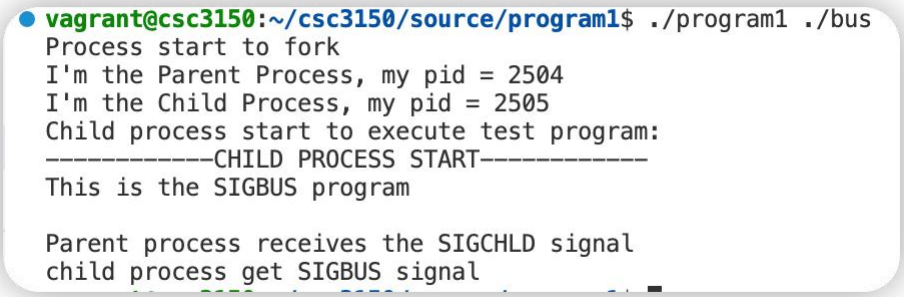
Test of abort:



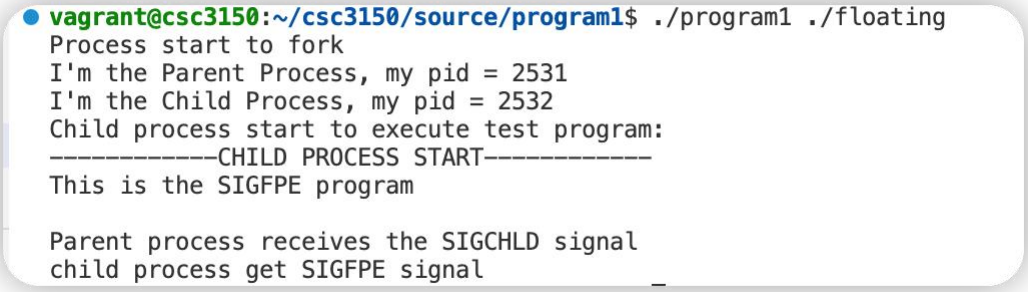
Test of alarm:



Test of bus:



Test of floating:



Test of hang up:

```
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./hangup
Process start to fork
I'm the Parent Process, my pid = 2570
I'm the Child Process, my pid = 2571
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGHUP program

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

Test of illegal_instr:

```
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./illegal_instr
Process start to fork
I'm the Parent Process, my pid = 2617
I'm the Child Process, my pid = 2618
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGILL program

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

Test of interrupt:

```
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./interrupt
Process start to fork
I'm the Parent Process, my pid = 2721
I'm the Child Process, my pid = 2722
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGINT program

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

Test of kill:

```
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./kill
Process start to fork
I'm the Parent Process, my pid = 2759
I'm the Child Process, my pid = 2760
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGKILL program

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

Test of normal:

```
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./normal
Process start to fork
I'm the Parent Process, my pid = 2797
I'm the Child Process, my pid = 2798
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
```

Test of pipe:

```
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./pipe
Process start to fork
I'm the Parent Process, my pid = 2823
I'm the Child Process, my pid = 2824
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGPIPE program

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

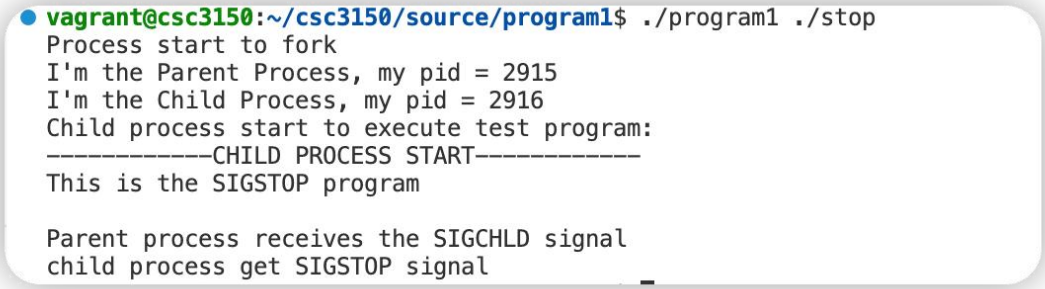
Test of quit:

```
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./quit
Process start to fork
I'm the Parent Process, my pid = 2849
I'm the Child Process, my pid = 2850
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGQUIT program

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

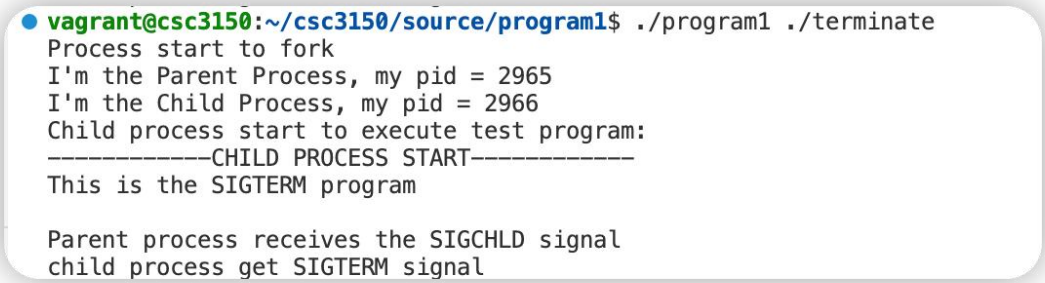
Test of segment_fault:

```
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./segment_fault
Process start to fork
I'm the Parent Process, my pid = 2888
I'm the Child Process, my pid = 2889
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGSEGV program

Parent process receives the SIGCHLD signal
child process get SIGEGV signal
```

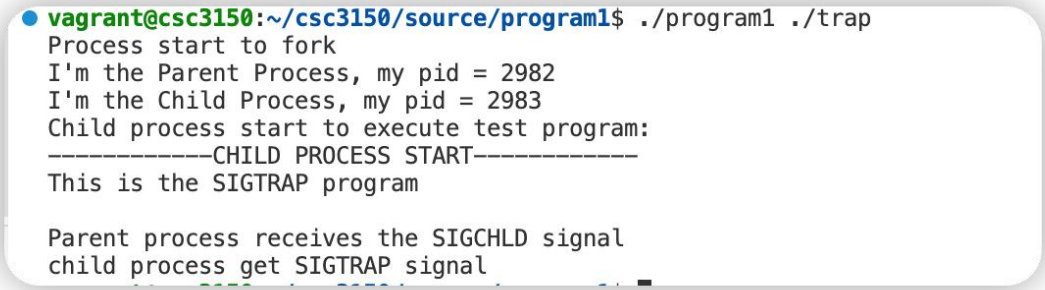
Test of stop:



Test of terminate:



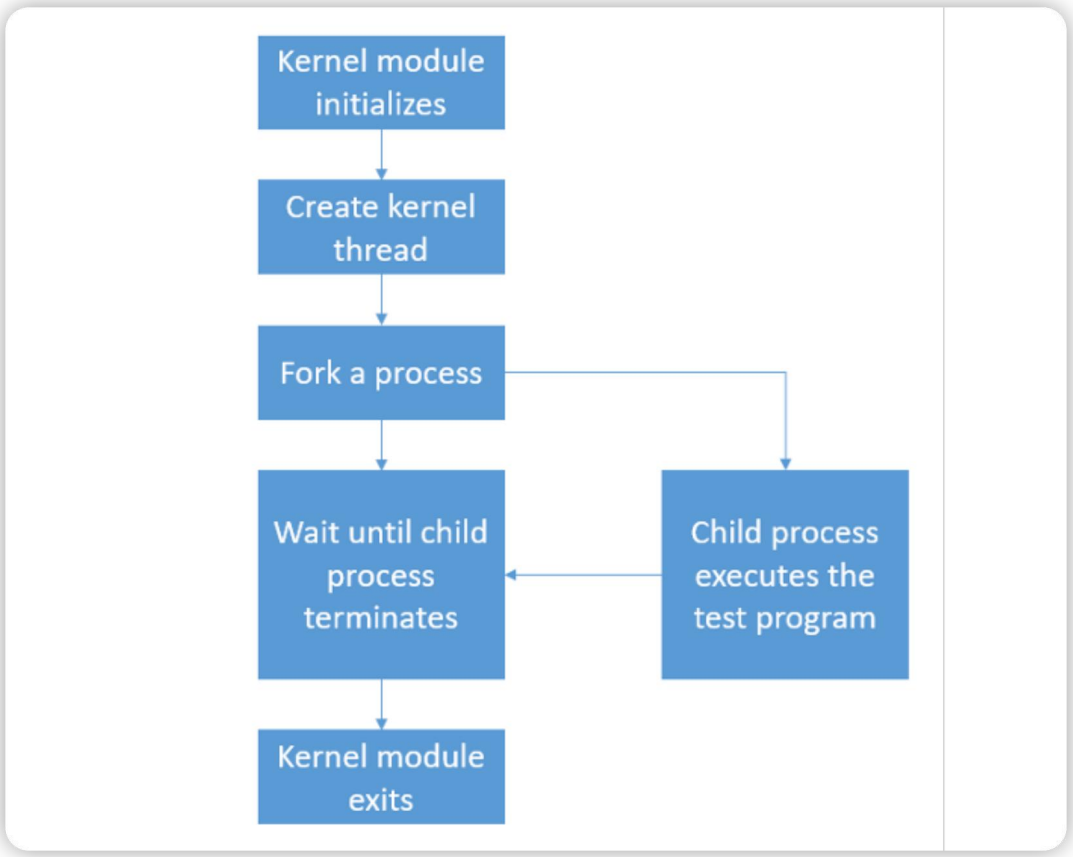
Test of trap:



What I learned from the project:
First I learned how to create the child process in the user mode. I also learned how to execute the test program in the child process and how to raise and receive signal between the child process and the parent process. Finally, I learned the different signals` basic knowledge.

Task2
Program idea:
This program is done in the kernel mode. Because I need the kernel_thread, do_execve, getname_kernel and do_wait these basic functions, I read the original code of the implementation of them in the linux system and use EXPORT_SYMBOL(function) to export them out so that I can use them. After the modification of the kernel, I can use extern to use them. Then I use the kthread_create and my_fork function to create the new child process. To implement the my_fork, I need to implement two advanced function called my_exec and my_wait. My_exec function will use getname_kernel to get

the code of the specific path (the test program) and use do_execve() to execute the test file. My_wait will use the do_wait to let parent process receives the signals from the child process. In my_fork(), after the parent receives the signals, it will analyze the signal and output for the signal.
The theoretical process is shown below:



Environment:
basic linux environment:
Linux Distribution: Ubuntu 20.04
Linux Kernel Version: 5.10.5
GCC Version: 5.4.0
Besides these:
1. Use EXPORT_SYMBOL() to export the function of fours.
2. Extern the function in program2.c.
3. Use below command to compile the kernel:
   make bzImage; make modules; make modules_install; make install; reboot
4. Type “make” .
5. Type “gcc test.c -o test”.
6. Type “insmod program2.ko” and “rmmod program2.ko”.
7. Using dmesg to display the message.

Screenshot:
Original test:

```
[ 2255.838243] [program2] : module_init {Wang Shijie} {120090331}
[ 2255.838245] [program2] : module_init create kthread start
[ 2255.838353] [program2] : module_init kthread start
[ 2255.838394] [program2] : The child process has pid = 4211
[ 2255.838396] [program2] : This is the parent process, pid = 4210
[ 2255.838442] [program2] : child process
[ 2255.939170] [program2] : get SIGBUS signal
[ 2255.939171] [program2] : child process has bus error
[ 2255.939172] [program2] : The return signal is 7
[ 2262.468840] [program2] : module_exit./my
```

Test abort:

```
[ 2454.461778] [program2] : module_init {Wang Shijie} {120090331}
[ 2454.461780] [program2] : module_init create kthread start
[ 2454.461896] [program2] : module_init kthread start
[ 2454.461944] [program2] : The child process has pid = 4451
[ 2454.461947] [program2] : This is the parent process, pid = 4450
[ 2454.461991] [program2] : child process
[ 2454.561670] [program2] : get SIGABRT signal
[ 2454.561672] [program2] : child process has abort error
[ 2454.561672] [program2] : The return signal is 6
[ 2462.208790] [program2] : module_exit./my
```

Test alarm:

```
[ 2522.974005] [program2] : module_init {Wang Shijie} {120090331}
[ 2522.974007] [program2] : module_init create kthread start
[ 2522.974092] [program2] : module_init kthread start
[ 2522.974140] [program2] : The child process has pid = 4654
[ 2522.974141] [program2] : This is the parent process, pid = 4653
[ 2522.974285] [program2] : child process
[ 2524.975954] [program2] : get SIGALRM signal
[ 2524.975956] [program2] : child process has alarm error
[ 2524.975956] [program2] : The return signal is 14
[ 2529.780120] [program2] : module_exit./my
```

Test bus:

```
[ 2592.866854] [program2] : module_init {Wang Shijie} {120090331}
[ 2592.866856] [program2] : module_init create kthread start
[ 2592.866900] [program2] : module_init kthread start
[ 2592.866931] [program2] : The child process has pid = 4792
[ 2592.866932] [program2] : This is the parent process, pid = 4791
[ 2592.867092] [program2] : child process
[ 2592.963642] [program2] : get SIGBUS signal
[ 2592.963643] [program2] : child process has bus error
[ 2592.963644] [program2] : The return signal is 7
[ 2597.592095] [program2] : module_exit./my
```

Test floating:

```
[ 2706.660817] [program2] : module_init {Wang Shijie} {120090331}
[ 2706.660819] [program2] : module_init create kthread start
[ 2706.660938] [program2] : module_init kthread start
[ 2706.660987] [program2] : The child process has pid = 5050
[ 2706.660989] [program2] : This is the parent process, pid = 5049
[ 2706.661045] [program2] : child process
[ 2706.762331] [program2] : get SIGFPE signal
[ 2706.762332] [program2] : child process has floating error
[ 2706.762333] [program2] : The return signal is 8
[ 2711.457871] [program2] : module_exit./my
```

Test hangup:

```
[ 2761.527484] [program2] : module_init {Wang Shijie} {120090331}
[ 2761.527486] [program2] : module_init create kthread start
[ 2761.527560] [program2] : module_init kthread start
[ 2761.527625] [program2] : The child process has pid = 5189
[ 2761.527627] [program2] : This is the parent process, pid = 5188
[ 2761.527676] [program2] : child process
[ 2761.528232] [program2] : get SIGHUP signal
[ 2761.528233] [program2] : child process hung up
[ 2761.528234] [program2] : The return signal is 1
[ 2770.324698] [program2] : module_exit./my
```

Test illegal_instr:

```
[ 2870.730424] [program2] : module_init {Wang Shijie} {120090331}
[ 2870.730426] [program2] : module_init create kthread start
[ 2870.730545] [program2] : module_init kthread start
[ 2870.730670] [program2] : The child process has pid = 5327
[ 2870.730672] [program2] : This is the parent process, pid = 5326
[ 2870.730693] [program2] : child process
[ 2870.833046] [program2] : get SIGILL signal
[ 2870.833048] [program2] : child process has illegal_instr error
[ 2870.833048] [program2] : The return signal is 4
[ 2874.606560] [program2] : module_exit./my
```

Test interrupt:

```
[ 3721.221359] [program2] : module_init {Wang Shijie} {120090331}
[ 3721.221362] [program2] : module_init create kthread start
[ 3721.221498] [program2] : module_init kthread start
[ 3721.221540] [program2] : The child process has pid = 5544
[ 3721.221541] [program2] : This is the parent process, pid = 5543
[ 3721.221645] [program2] : child process
[ 3721.222049] [program2] : get SIGINT signal
[ 3721.222050] [program2] : terminal interrupt
[ 3721.222051] [program2] : The return signal is 2
[ 4347.459245] [program2] : module_exit./my
```

Test kill:

```
[ 5153.334828] [program2] : module_init {Wang Shijie} {120090331}
[ 5153.334830] [program2] : module_init create kthread start
[ 5153.334932] [program2] : module_init kthread start
[ 5153.335058] [program2] : The child process has pid = 5983
[ 5153.335060] [program2] : This is the parent process, pid = 5982
[ 5153.335114] [program2] : child process
[ 5153.335922] [program2] : get SIGKILL signal
[ 5153.335923] [program2] : child process killed
[ 5153.335924] [program2] : The return signal is 9
[ 5160.317723] [program2] : module_exit./my
```

Test normal:

```
[ 5160.317723] [program2] : module_exit./my
[ 5238.665737] [program2] : module_init {Wang Shijie} {120090331}
[ 5238.665739] [program2] : module_init create kthread start
[ 5238.665840] [program2] : module_init kthread start
[ 5238.665885] [program2] : The child process has pid = 6183
[ 5238.665886] [program2] : This is the parent process, pid = 6182
[ 5238.665927] [program2] : child process
[ 5238.666544] [program2] : child process gets normal termination
[ 5238.666546] [program2] : The return signal is 0
[ 5243.720158] [program2] : module_exit./my
```

Test pipe:

```
[ 5297.832220] [program2] : module_init {Wang Shijie} {120090331}
[ 5297.832222] [program2] : module_init create kthread start
[ 5297.832340] [program2] : module_init kthread start
[ 5297.832428] [program2] : The child process has pid = 6299
[ 5297.832430] [program2] : This is the parent process, pid = 6298
[ 5297.832446] [program2] : child process
[ 5297.832925] [program2] : get SIGPIPE signal
[ 5297.832927] [program2] : child process has pipe error
[ 5297.832927] [program2] : The return signal is 13
[ 5303.184739] [program2] : module_exit./my
```

Test quit:

```
[ 5363.460589] [program2] : module_init {Wang Shijie} {120090331}
[ 5363.460591] [program2] : module_init create kthread start
[ 5363.460689] [program2] : module_init kthread start
[ 5363.460753] [program2] : The child process has pid = 6481
[ 5363.460754] [program2] : This is the parent process, pid = 6480
[ 5363.460758] [program2] : child process
[ 5363.563311] [program2] : get SIGQUIT signal
[ 5363.563313] [program2] : terminal quit
[ 5363.563313] [program2] : The return signal is 3
[ 5367.415130] [program2] : module_exit./my
```

Test segmentation fault:

```
[ 5413.789387] [program2] : module_init {Wang Shijie} {120090331}
[ 5413.789408] [program2] : module_init create kthread start
[ 5413.789508] [program2] : module_init kthread start
[ 5413.789956] [program2] : The child process has pid = 6656
[ 5413.789957] [program2] : This is the parent process, pid = 6654
[ 5413.790017] [program2] : child process
[ 5413.897017] [program2] : get SIGSEGV signal
[ 5413.897019] [program2] : child process has segmentation fault
[ 5413.897020] [program2] : The return signal is 11
[ 5418.452409] [program2] : module_exit./my
```

Test stop:

```
[ 5517.667277] [program2] : module_init {Wang Shijie} {120090331}
[ 5517.667279] [program2] : module_init create kthread start
[ 5517.667454] [program2] : module_init kthread start
[ 5517.667519] [program2] : The child process has pid = 6828
[ 5517.667521] [program2] : This is the parent process, pid = 6827
[ 5517.667580] [program2] : child process
[ 5517.668106] [program2] : Child process stopped
[ 5517.668108] [program2] : The return signal is 19
[ 5521.765375] [program2] : module_exit./my
```

Test terminate:

```
[ 5561.862361] [program2] : module_init {Wang Shijie} {120090331}
[ 5561.862363] [program2] : module_init create kthread start
[ 5561.862506] [program2] : module_init kthread start
[ 5561.862602] [program2] : The child process has pid = 6974
[ 5561.862603] [program2] : This is the parent process, pid = 6973
[ 5561.862621] [program2] : child process
[ 5561.863375] [program2] : get SIGTERM signal
[ 5561.863376] [program2] : child process terminated
[ 5561.863377] [program2] : The return signal is 15
[ 5565.626977] [program2] : module_exit./my
```

Test trap:

```
[ 6232.183287] [program2] : module_init {Wang Shijie} {120090331}
[ 6232.183289] [program2] : module_init create kthread start
[ 6232.183409] [program2] : module_init kthread start
[ 6232.183458] [program2] : The child process has pid = 7194
[ 6232.183460] [program2] : This is the parent process, pid = 7193
[ 6232.183501] [program2] : child process
[ 6232.278315] [program2] : get SIGTRAP signal
[ 6232.278316] [program2] : child process has trap error
[ 6232.278317] [program2] : The return signal is 5
[ 6236.157823] [program2] : module_exit./my
```

What I have learned from the project:

In this project, I have learned how to modify the kernel and recompile the kernel by LKM. What`s more, I also learned how to insert and remove the modules to the kernels and show the message of the kernel.