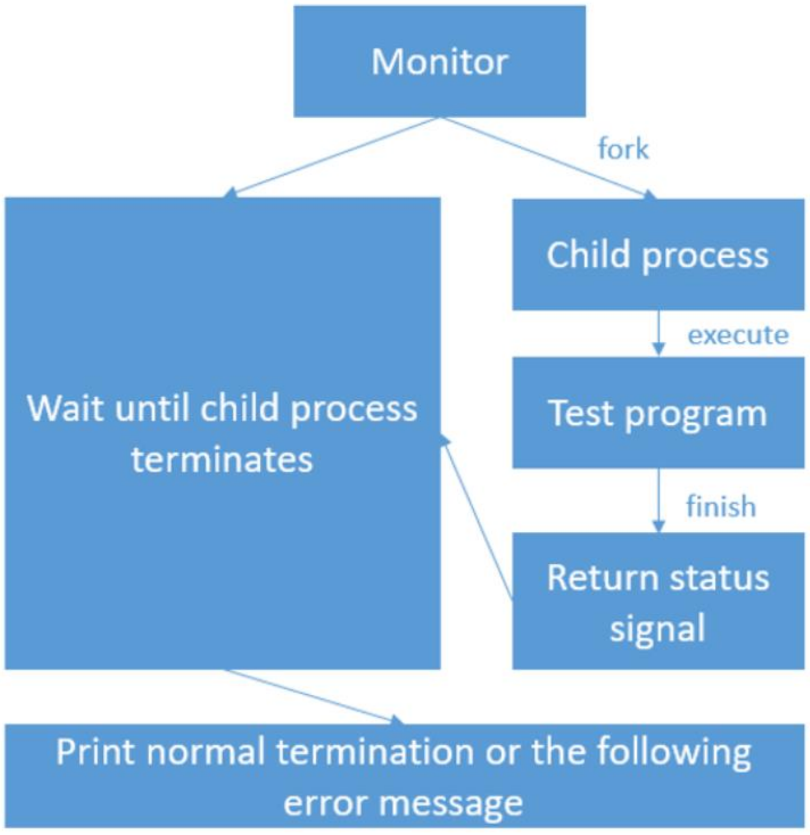
CSC3150 Assignment 1 Report

120090525

Xingjian Li

Task1：

Design:


Monitor
fork
Child process
execute
Test program
finish
Return status signal
Wait until child process terminates
Print normal termination or the following error message

The design of task 1 is mainly based on the figure above. It can be roughly divided into two parts: one is forking a child process and the other one is analyzing the return signal of child process.

First, I use fork to get the child process. After child process finish execution, we will get the signal of the child process. Finally we use WIFEXITED, WIFSIGNALED, and WIFSTOPPED to decide the return status, and then I can print the status of child process.

Set up environment:

Follow the tut to compile the kernel.

HOW TO COMPILE:

    In the 'program1' directory, type 'make' command and enter.

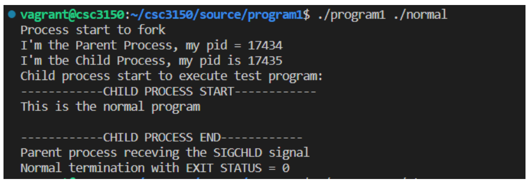
HOW TO CLEAR:

    In the 'program1' directory, type 'make clean' command and enter.

HOW TO EXECUTE:

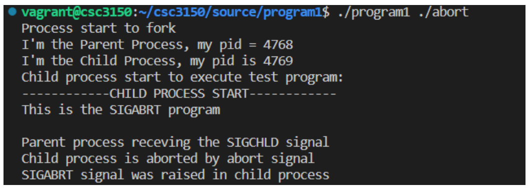
    In the 'program1' directory, type './program1 $TEST_CASE $ARG1 $ARG2 ...',
    where $TEST_CASE is the name of test program and $ARG1, $ARG2,...
    are names of arguments that the test program could have.

Result:



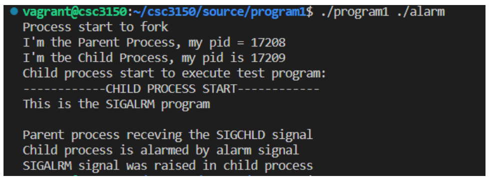
```
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./normal
Process start to fork
I'm the Parent Process, my pid = 17434
I'm tbe Child Process, my pid is 17435
Child process start to execute test program:
-----------CHILD PROCESS START-----------
This is the normal program

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



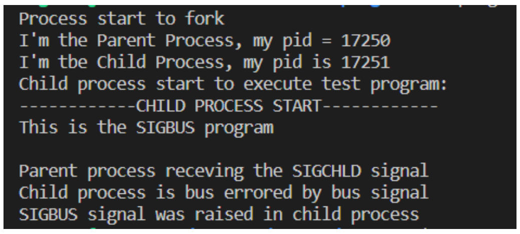
```
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./abort
Process start to fork
I'm the Parent Process, my pid = 4768
I'm tbe Child Process, my pid is 4769
Child process start to execute test program:
-----------CHILD PROCESS START-----------
This is the SIGABRT program

Parent process receving the SIGCHLD signal
Child process is aborted by abort signal
SIGABRT signal was raised in child process
```



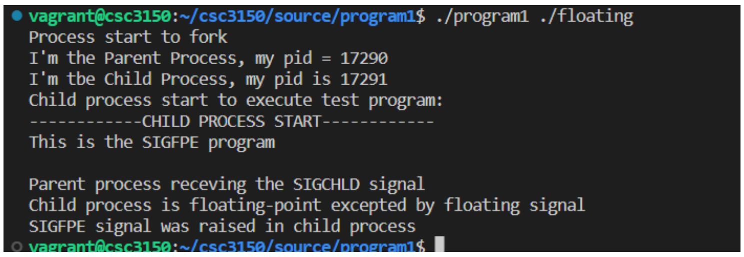
```
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./alarm
Process start to fork
I'm the Parent Process, my pid = 17208
I'm tbe Child Process, my pid is 17209
Child process start to execute test program:
-----------CHILD PROCESS START-----------
This is the SIGALRM program

Parent process receving the SIGCHLD signal
Child process is alarmed by alarm signal
SIGALRM signal was raised in child process
```



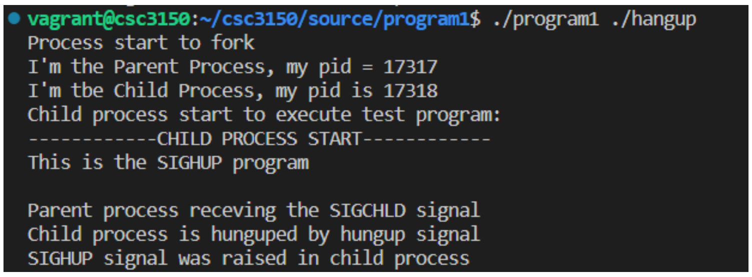
```
Process start to fork
I'm the Parent Process, my pid = 17250
I'm tbe Child Process, my pid is 17251
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGBUS program

Parent process receving the SIGCHLD signal
Child process is bus errored by bus signal
SIGBUS signal was raised in child process
```



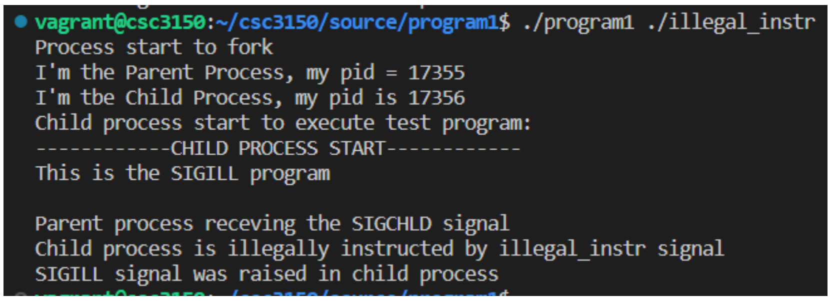
```
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./floating
Process start to fork
I'm the Parent Process, my pid = 17290
I'm tbe Child Process, my pid is 17291
Child process start to execute test program:
-----------CHILD PROCESS START-----------
This is the SIGFPE program

Parent process receving the SIGCHLD signal
Child process is floating-point excepted by floating signal
SIGFPE signal was raised in child process
vagrant@csc3150:~/csc3150/source/program1$
```



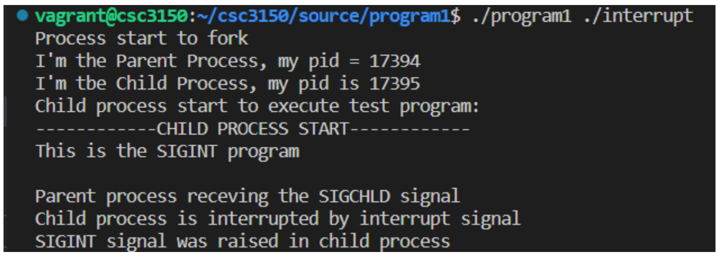
```
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./hangup
Process start to fork
I'm the Parent Process, my pid = 17317
I'm tbe Child Process, my pid is 17318
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGHUP program

Parent process receving the SIGCHLD signal
Child process is hunguped by hungup signal
SIGHUP signal was raised in child process
```



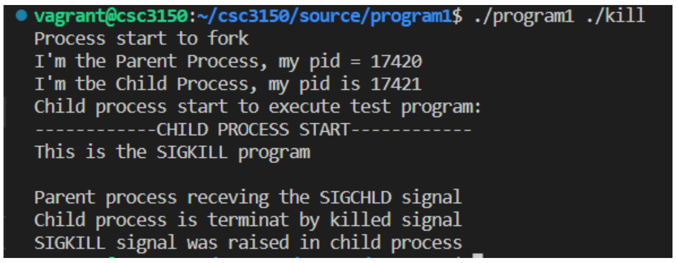
```
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./illegal_instr
Process start to fork
I'm the Parent Process, my pid = 17355
I'm tbe Child Process, my pid is 17356
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGILL program

Parent process receving the SIGCHLD signal
Child process is illegally instructed by illegal_instr signal
SIGILL signal was raised in child process
```



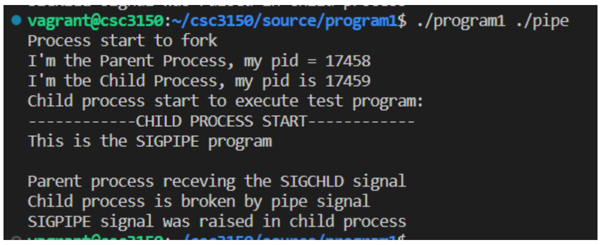
```
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./interrupt
Process start to fork
I'm the Parent Process, my pid = 17394
I'm tbe Child Process, my pid is 17395
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGINT program

Parent process receving the SIGCHLD signal
Child process is interrupted by interrupt signal
SIGINT signal was raised in child process
```



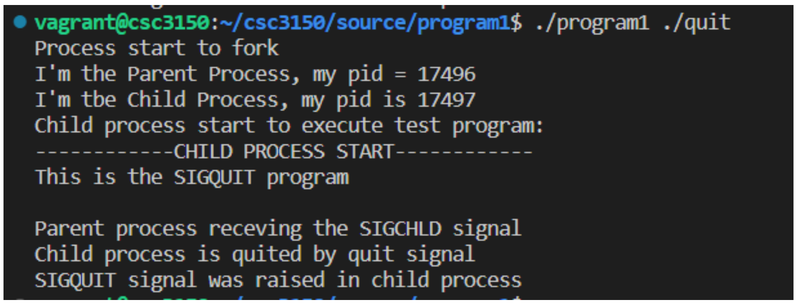
```
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./kill
Process start to fork
I'm the Parent Process, my pid = 17420
I'm tbe Child Process, my pid is 17421
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGKILL program

Parent process receving the SIGCHLD signal
Child process is terminat by killed signal
SIGKILL signal was raised in child process
```



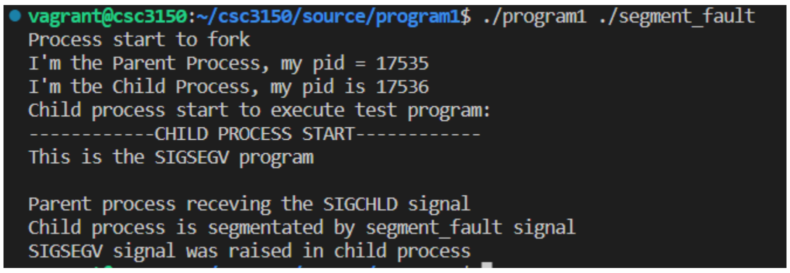
```
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./pipe
Process start to fork
I'm the Parent Process, my pid = 17458
I'm tbe Child Process, my pid is 17459
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGPIPE program

Parent process receving the SIGCHLD signal
Child process is broken by pipe signal
SIGPIPE signal was raised in child process
```



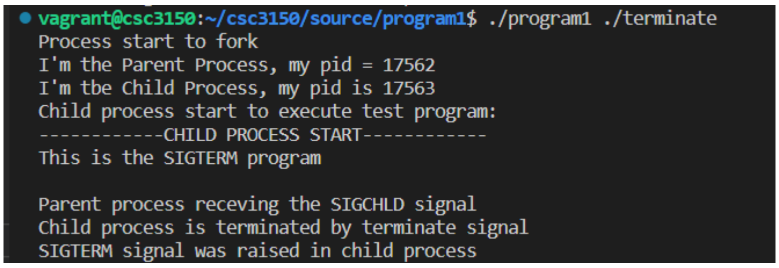
```
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./quit
Process start to fork
I'm the Parent Process, my pid = 17496
I'm tbe Child Process, my pid is 17497
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGQUIT program

Parent process receving the SIGCHLD signal
Child process is quited by quit signal
SIGQUIT signal was raised in child process
```



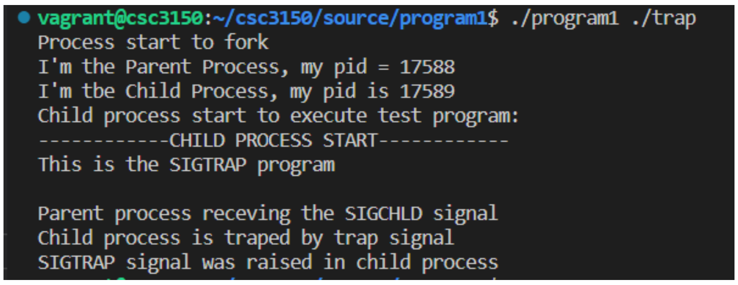
```
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./segment_fault
Process start to fork
I'm the Parent Process, my pid = 17535
I'm tbe Child Process, my pid is 17536
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGSEGV program

Parent process receving the SIGCHLD signal
Child process is segmentated by segment_fault signal
SIGSEGV signal was raised in child process
```



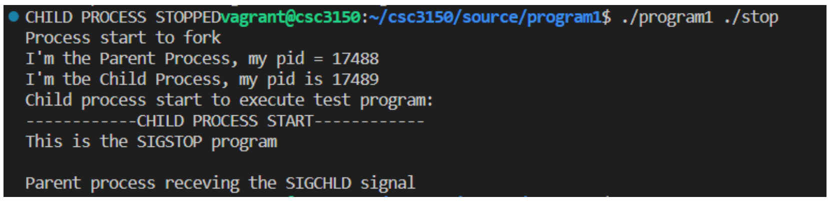
```
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./terminate
Process start to fork
I'm the Parent Process, my pid = 17562
I'm tbe Child Process, my pid is 17563
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGTERM program

Parent process receving the SIGCHLD signal
Child process is terminated by terminate signal
SIGTERM signal was raised in child process
```



```
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./trap
Process start to fork
I'm the Parent Process, my pid = 17588
I'm tbe Child Process, my pid is 17589
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGTRAP program

Parent process receving the SIGCHLD signal
Child process is traped by trap signal
SIGTRAP signal was raised in child process
```



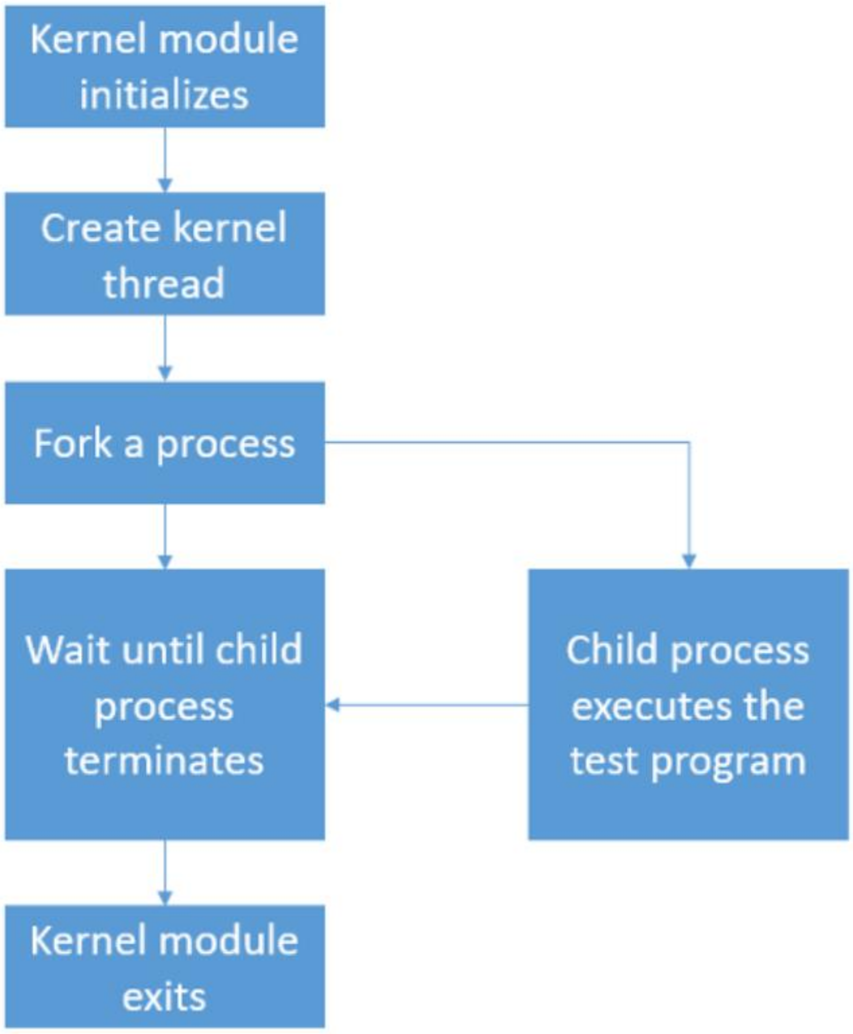
```
CHILD PROCESS STOPPEDvagrant@csc3150:~/csc3150/source/program1$ ./program1 ./stop
Process start to fork
I'm the Parent Process, my pid = 17488
I'm tbe Child Process, my pid is 17489
Child process start to execute test program:
-----------CHILD PROCESS START-----------
This is the SIGSTOP program

Parent process receving the SIGCHLD signal
```

Task2:

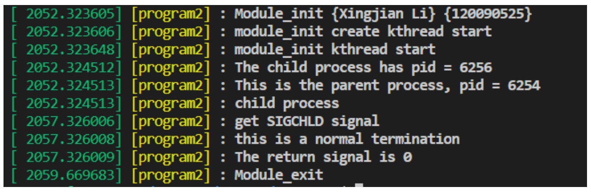
Design

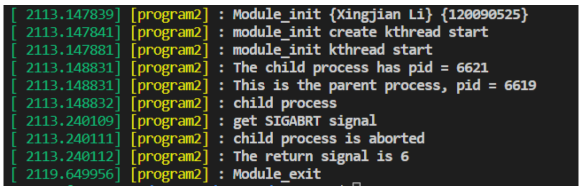
The main flow chart for Task 2 is:

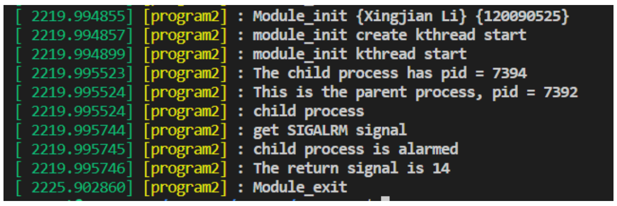

Kernel module initializes
Create kernel thread
Fork a process
Wait until child process terminates
Child process executes the test program
Kernel module exits

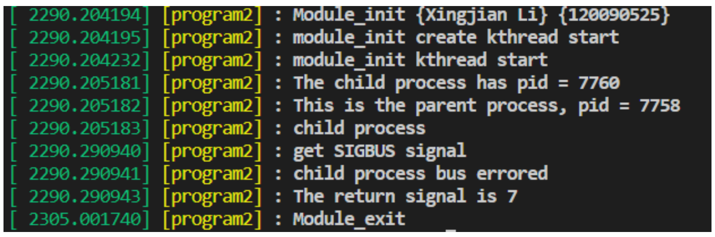
It is same to task1, but we need to do it in the kernel mode. First forking a child process, then make parent process waits until child process terminates and evaluate child process's

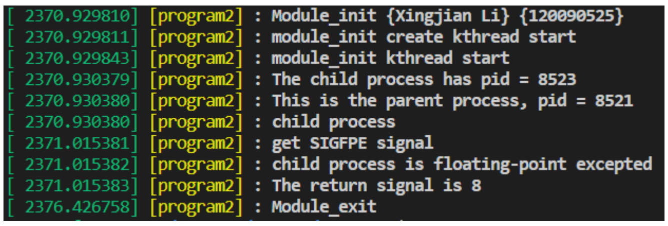
exact signal argument.
Set up environment:
BEFORE PROGRAM COMPILATION AND EXECUTION:
Revising Linux Kernel is needed, as is shown in the following steps:
1. Update the Linux source code.(include do_wait, do_execve, kernel_clone, getname_kernel) use EXPORT_SYMBOL
2. Compile the kernel and boot image, replace the boot image with new one, then reboot.
HOW TO COMPILE:
In the 'program2' directory, type 'make' command and enter
HOW TO CLEAR:
In the 'program2' directory, type 'make clean' command and enter.
HOW TO EXECUTE:
1.Type 'sudo insmod program2.ko' under 'program2' directory and enter
2.Type 'sudo rmmod program2' and enter to remove the program2 module.
3.You could see messages appear by typing 'dmesg' command. The messages are between the messages 'module init' and 'module exit'.
Result:

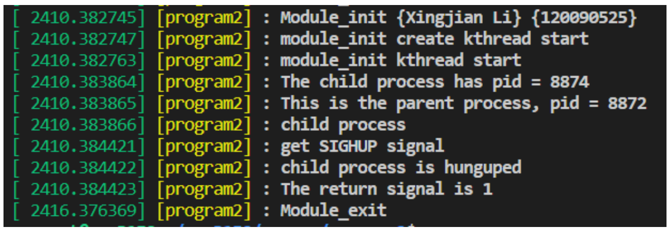

```
[ 2052.323605] [program2] : Module_init {Xingjian Li} {120090525}
[ 2052.323606] [program2] : module_init create kthread start
[ 2052.323648] [program2] : module_init kthread start
[ 2052.324512] [program2] : The child process has pid = 6256
[ 2052.324513] [program2] : This is the parent process, pid = 6254
[ 2052.324513] [program2] : child process
[ 2057.326006] [program2] : get SIGCHLD signal
[ 2057.326008] [program2] : this is a normal termination
[ 2057.326009] [program2] : The return signal is 0
[ 2059.669683] [program2] : Module_exit
```


```
[ 2113.147839] [program2] : Module_init {Xingjian Li} {120090525}
[ 2113.147841] [program2] : module_init create kthread start
[ 2113.147881] [program2] : module_init kthread start
[ 2113.148831] [program2] : The child process has pid = 6621
[ 2113.148831] [program2] : This is the parent process, pid = 6619
[ 2113.148832] [program2] : child process
[ 2113.240109] [program2] : get SIGABRT signal
[ 2113.240111] [program2] : child process is aborted
[ 2113.240112] [program2] : The return signal is 6
[ 2119.649956] [program2] : Module_exit
```

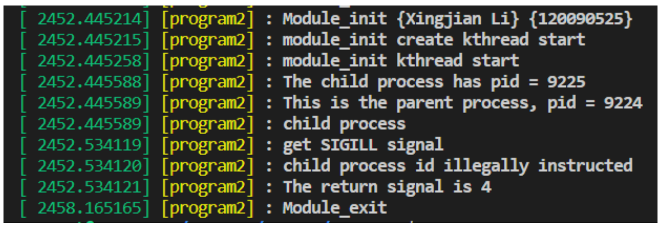

```
[ 2219.994855] [program2] : Module_init {Xingjian Li} {120090525}
[ 2219.994857] [program2] : module_init create kthread start
[ 2219.994899] [program2] : module_init kthread start
[ 2219.995523] [program2] : The child process has pid = 7394
[ 2219.995524] [program2] : This is the parent process, pid = 7392
[ 2219.995524] [program2] : child process
[ 2219.995744] [program2] : get SIGALRM signal
[ 2219.995745] [program2] : child process is alarmed
[ 2219.995746] [program2] : The return signal is 14
[ 2225.902860] [program2] : Module_exit
```


```
[ 2290.204194] [program2] : Module_init {Xingjian Li} {120090525}
[ 2290.204195] [program2] : module_init create kthread start
[ 2290.204232] [program2] : module_init kthread start
[ 2290.205181] [program2] : The child process has pid = 7760
[ 2290.205182] [program2] : This is the parent process, pid = 7758
[ 2290.205183] [program2] : child process
[ 2290.290940] [program2] : get SIGBUS signal
[ 2290.290941] [program2] : child process bus errored
[ 2290.290943] [program2] : The return signal is 7
[ 2305.001740] [program2] : Module_exit
```

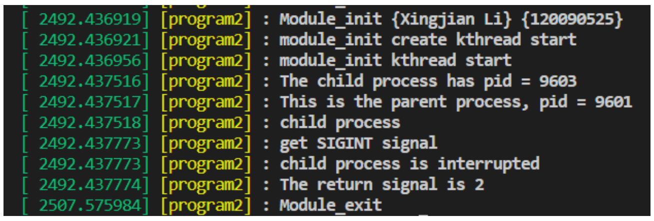

```
[ 2370.929810] [program2] : Module_init {Xingjian Li} {120090525}
[ 2370.929811] [program2] : module_init create kthread start
[ 2370.929843] [program2] : module_init kthread start
[ 2370.930379] [program2] : The child process has pid = 8523
[ 2370.930380] [program2] : This is the parent process, pid = 8521
[ 2370.930380] [program2] : child process
[ 2371.015381] [program2] : get SIGFPE signal
[ 2371.015382] [program2] : child process is floating-point excepted
[ 2371.015383] [program2] : The return signal is 8
[ 2376.426758] [program2] : Module_exit
```



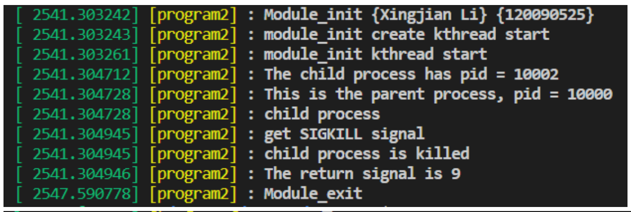
```
[ 2410.382745] [program2] : Module_init {Xingjian Li} {120090525}
[ 2410.382747] [program2] : module_init create kthread start
[ 2410.382763] [program2] : module_init kthread start
[ 2410.383864] [program2] : The child process has pid = 8874
[ 2410.383865] [program2] : This is the parent process, pid = 8872
[ 2410.383866] [program2] : child process
[ 2410.384421] [program2] : get SIGHUP signal
[ 2410.384422] [program2] : child process is hunguped
[ 2410.384423] [program2] : The return signal is 1
[ 2416.376369] [program2] : Module_exit
```



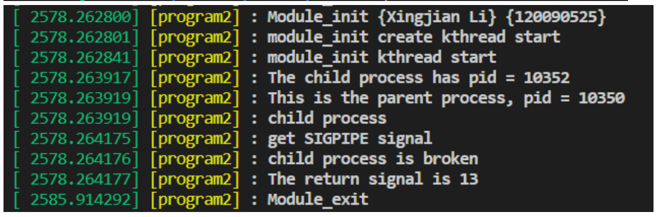
```
[ 2452.445214] [program2] : Module_init {Xingjian Li} {120090525}
[ 2452.445215] [program2] : module_init create kthread start
[ 2452.445258] [program2] : module_init kthread start
[ 2452.445588] [program2] : The child process has pid = 9225
[ 2452.445589] [program2] : This is the parent process, pid = 9224
[ 2452.445589] [program2] : child process
[ 2452.534119] [program2] : get SIGILL signal
[ 2452.534120] [program2] : child process id illegally instructed
[ 2452.534121] [program2] : The return signal is 4
[ 2458.165165] [program2] : Module_exit
```



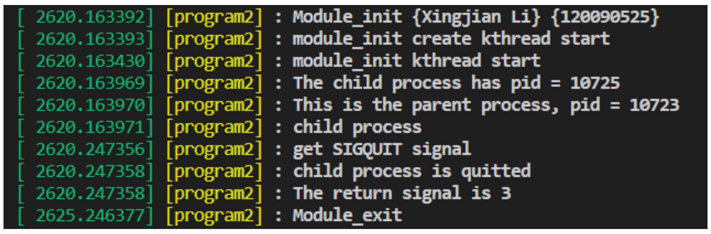
```
[ 2492.436919] [program2] : Module_init {Xingjian Li} {120090525}
[ 2492.436921] [program2] : module_init create kthread start
[ 2492.436956] [program2] : module_init kthread start
[ 2492.437516] [program2] : The child process has pid = 9603
[ 2492.437517] [program2] : This is the parent process, pid = 9601
[ 2492.437518] [program2] : child process
[ 2492.437773] [program2] : get SIGINT signal
[ 2492.437773] [program2] : child process is interrupted
[ 2492.437774] [program2] : The return signal is 2
[ 2507.575984] [program2] : Module_exit
```



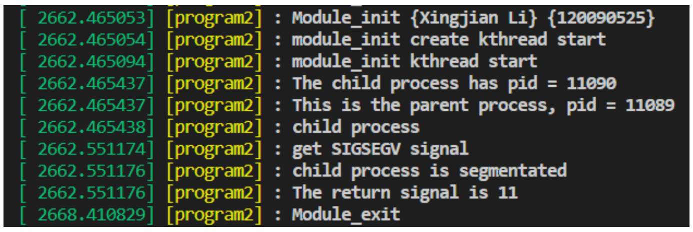
```
[ 2541.303242] [program2] : Module_init {Xingjian Li} {120090525}
[ 2541.303243] [program2] : module_init create kthread start
[ 2541.303261] [program2] : module_init kthread start
[ 2541.304712] [program2] : The child process has pid = 10002
[ 2541.304728] [program2] : This is the parent process, pid = 10000
[ 2541.304728] [program2] : child process
[ 2541.304945] [program2] : get SIGKILL signal
[ 2541.304945] [program2] : child process is killed
[ 2541.304946] [program2] : The return signal is 9
[ 2547.590778] [program2] : Module_exit
```



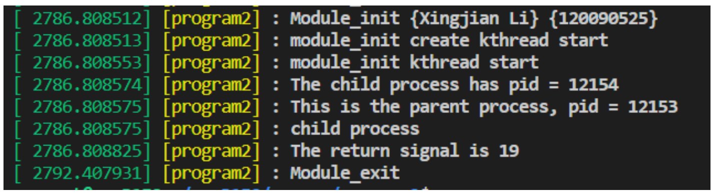
```
[ 2578.262800] [program2] : Module_init {Xingjian Li} {120090525}
[ 2578.262801] [program2] : module_init create kthread start
[ 2578.262841] [program2] : module_init kthread start
[ 2578.263917] [program2] : The child process has pid = 10352
[ 2578.263919] [program2] : This is the parent process, pid = 10350
[ 2578.263919] [program2] : child process
[ 2578.264175] [program2] : get SIGPIPE signal
[ 2578.264176] [program2] : child process is broken
[ 2578.264177] [program2] : The return signal is 13
[ 2585.914292] [program2] : Module_exit
```



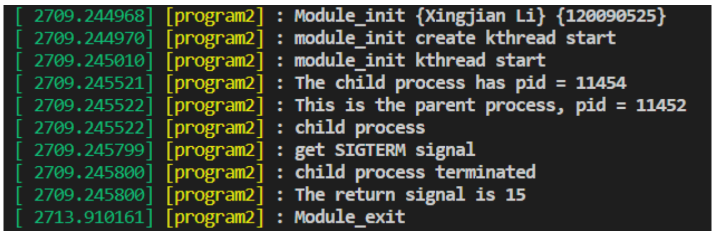
```
[ 2620.163392] [program2] : Module_init {Xingjian Li} {120090525}
[ 2620.163393] [program2] : module_init create kthread start
[ 2620.163430] [program2] : module_init kthread start
[ 2620.163969] [program2] : The child process has pid = 10725
[ 2620.163970] [program2] : This is the parent process, pid = 10723
[ 2620.163971] [program2] : child process
[ 2620.247356] [program2] : get SIGQUIT signal
[ 2620.247358] [program2] : child process is quitted
[ 2620.247358] [program2] : The return signal is 3
[ 2625.246377] [program2] : Module_exit
```



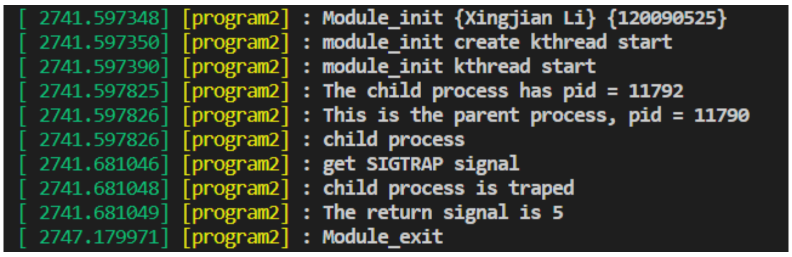
```
[ 2662.465053] [program2] : Module_init {Xingjian Li} {120090525}
[ 2662.465054] [program2] : module_init create kthread start
[ 2662.465094] [program2] : module_init kthread start
[ 2662.465437] [program2] : The child process has pid = 11090
[ 2662.465437] [program2] : This is the parent process, pid = 11089
[ 2662.465438] [program2] : child process
[ 2662.551174] [program2] : get SIGSEGV signal
[ 2662.551176] [program2] : child process is segmentated
[ 2662.551176] [program2] : The return signal is 11
[ 2668.410829] [program2] : Module_exit
```



```
[ 2786.808512] [program2] : Module_init {Xingjian Li} {120090525}
[ 2786.808513] [program2] : module_init create kthread start
[ 2786.808553] [program2] : module_init kthread start
[ 2786.808574] [program2] : The child process has pid = 12154
[ 2786.808575] [program2] : This is the parent process, pid = 12153
[ 2786.808575] [program2] : child process
[ 2786.808825] [program2] : The return signal is 19
[ 2792.407931] [program2] : Module_exit
```



```
[ 2709.244968] [program2] : Module_init {Xingjian Li} {120090525}
[ 2709.244970] [program2] : module_init create kthread start
[ 2709.245010] [program2] : module_init kthread start
[ 2709.245521] [program2] : The child process has pid = 11454
[ 2709.245522] [program2] : This is the parent process, pid = 11452
[ 2709.245522] [program2] : child process
[ 2709.245799] [program2] : get SIGTERM signal
[ 2709.245800] [program2] : child process terminated
[ 2709.245800] [program2] : The return signal is 15
[ 2713.910161] [program2] : Module_exit
```



```
[ 2741.597348] [program2] : Module_init {Xingjian Li} {120090525}
[ 2741.597350] [program2] : module_init create kthread start
[ 2741.597390] [program2] : module_init kthread start
[ 2741.597825] [program2] : The child process has pid = 11792
[ 2741.597826] [program2] : This is the parent process, pid = 11790
[ 2741.597826] [program2] : child process
[ 2741.681046] [program2] : get SIGTRAP signal
[ 2741.681048] [program2] : child process is traped
[ 2741.681049] [program2] : The return signal is 5
[ 2747.179971] [program2] : Module_exit
```

Learning:

1. How to install, compile and use the kernel
2. How to use fork to get the child process
3. How to change the Linux source code