CSC3150 Assignment1

李翰豪
ID:120090657

October 10, 2022

1 Design

1.1 task 1

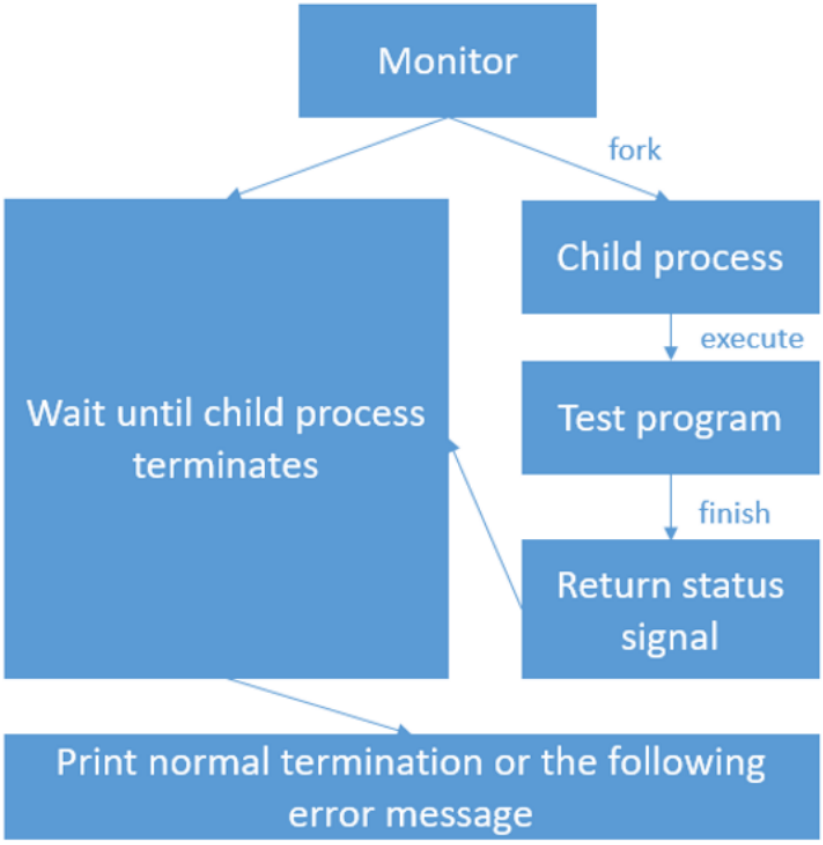

Figure 1: task1

In task1, I just follow the flow chart. Firstly, i use function fork() to open a child process. The return value of fork() can help me differentiate Parent process from child process. After creating child process, the Parent process only need to wait and receive the status signal from

the Child process. Since the test programs are provided to us, I just need to care about how to use function execve() correctly. The crucial part is how to output correct signal according to the signal status we have.

1.2 task 2

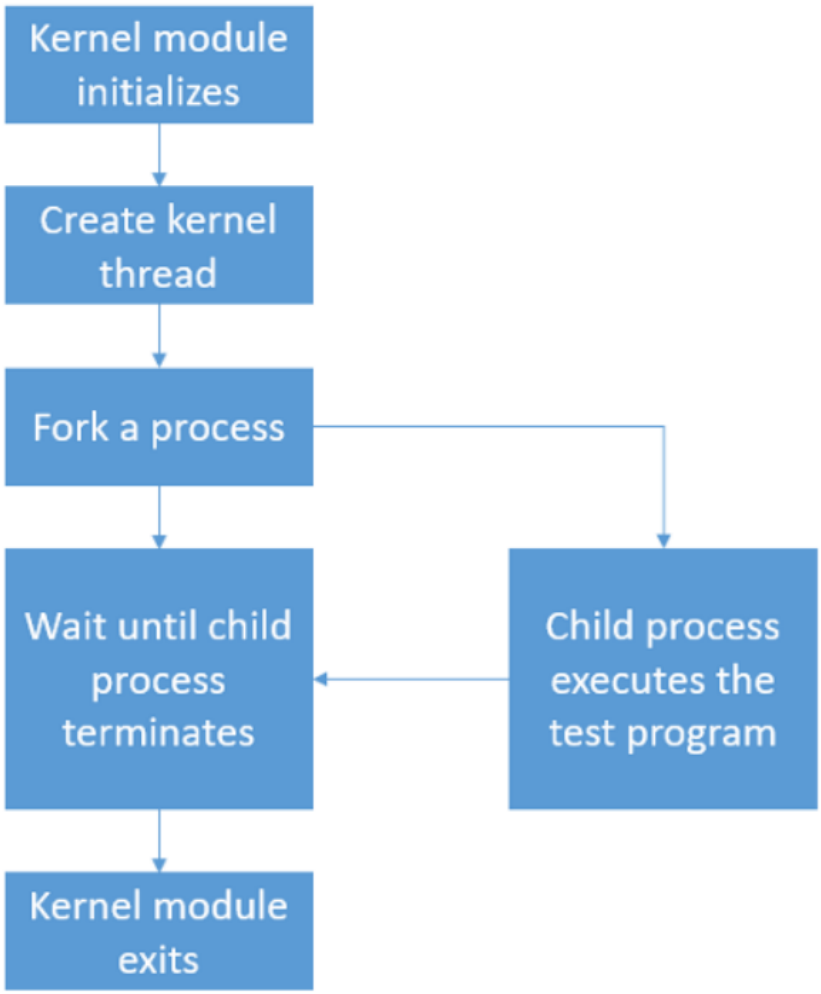

Figure 2: task2

Task2 is more troublesome than task 1. According to the flow chart, I can divide the program we need to implement into 4 parts. Firstly, how to create kernel thread. Secondly, how to fork a process in the thread just created. Thirdly, how to use interior function to construct my execution part. Fourthly, in the Parent process, how to implement the wait part. Since we have to use the functions inside the kernel, we have to go to the kernel and export them. After a million tries, i compile it successfully.

2 Environment Setting

I spend most of time in kernel compiling since i did not go to the tutorial onsite. It is too difficult and complicated to follow and understanding the whole process without any mistakes. There is one thing perplex me for three days. I only have no more than 5G in my Disk C. At first, I try to enlarge Disk C by empty Disk D which is near the Disk C, but i fail and almost ruin my computer. I search a lot of methods but just fail. Finally, i deleted a lot of games in my PC and many rubbish files manage to get enough space for my kernel. As for the rest staff, like GCC version or Code style, i just follow the tutorial and success easily.

3 Output Result

```
root@csc3150:/home/vagrant/csc3150/program1# ./program1 normal
Process start to fork
I'm the Parent Process, my pid = 15964
I'm the Child Process, my pid = 15965
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the normal program

------------CHILD PROCESS END------------
Parent process recieves SIGCHLD signal
Normal termination with EXIT STATUS = 0
root@csc3150:/home/vagrant/csc3150/program1# ./program1 abort
Process start to fork
I'm the Parent Process, my pid = 16051
I'm the Child Process, my pid = 16052
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGABRT program

Parent process recieves SIGCHLD signal
child process get SIGABRT signal
root@csc3150:/home/vagrant/csc3150/program1# ./program1 trap
Process start to fork
I'm the Parent Process, my pid = 16097
I'm the Child Process, my pid = 16098
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGTRAP program

Parent process recieves SIGCHLD signal
child process get SIGTRAP signal
root@csc3150:/home/vagrant/csc3150/program1# ./program1 stop
Process start to fork
I'm the Parent Process, my pid = 16227
I'm the Child Process, my pid = 16228
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGSTOP program

Parent process recieves SIGCHLD signal
child process get SIGSTOP signal
root@csc3150:/home/vagrant/csc3150/program1#
```

Figure 3: task1

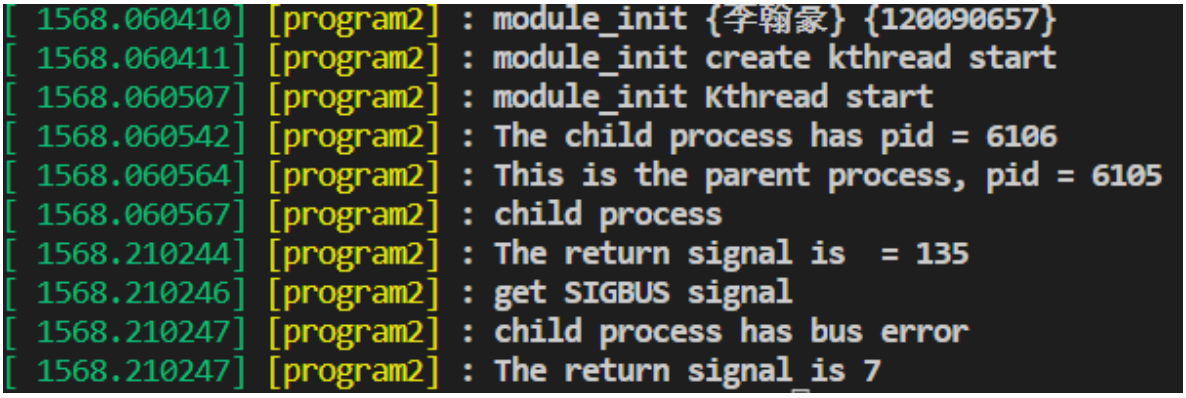

Figure 4: task2

4 Personal Feeling

I just find myself learning a lot from this process. I means something detach from the textbooks. For example, i have learn how to modify the source code from the kernel, by using vim! This assignment give me a chance to have a glimpse on the underlying code of the computer. I comprehend further about the operating system, like what the process do and how they do.