CSC3150 Assignment1 Report

LIU Hengrui
120090609

Task 1.

I design my program followed by the tutorial. First, we need to do nothing about the environment. I just combined the two function that shown in the tutorial and get the main structure of my program1. To show the right signal, I made a 2-dimensional array, which contains all the SIGCHILD signals. Here are my outputs:

```
root@csc3150:~/source/program1# ./program1 ./kill
Process start to fork
I'm the Parent Process, my pid = 30073
I'm the Child Process, my pid = 30074
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
```

```
root@csc3150:~/source/program1# ./program1 ./quit
Process start to fork
I'm the Parent Process, my pid = 30093
I'm the Child Process, my pid = 30094
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
```

```
root@csc3150:~/source/program1# ./program1 ./normal
Process start to fork
I'm the Parent Process, my pid = 30071
I'm the Child Process, my pid = 30072
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
```

```
root@csc3150:~/source/program1# ./program1 ./abort
Process start to fork
I'm the Parent Process, my pid = 30326
I'm the Child Process, my pid = 30327
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
```

```
root@csc3150:~/source/program1# ./program1 ./alarm
Process start to fork
I'm the Parent Process, my pid = 30383
I'm the Child Process, my pid = 30384
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
```

```
root@csc3150:~/source/program1# ./program1 ./bus
Process start to fork
I'm the Parent Process, my pid = 30425
I'm the Child Process, my pid = 30426
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
```

```
root@csc3150:~/source/program1# ./program1 ./floating
Process start to fork
I'm the Parent Process, my pid = 30465
I'm the Child Process, my pid = 30466
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
```

```
root@csc3150:~/source/program1# ./program1 ./hangup
Process start to fork
I'm the Parent Process, my pid = 30575
I'm the Child Process, my pid = 30576
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
```

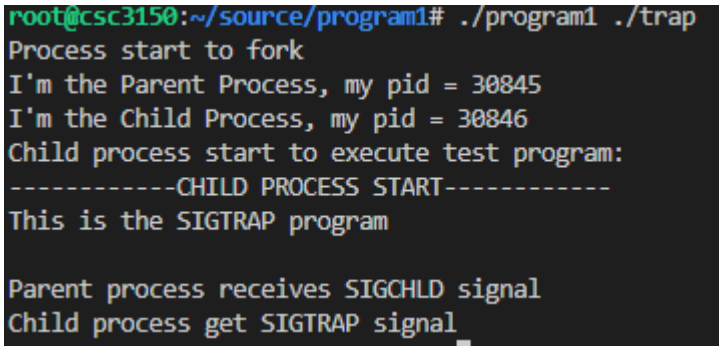
```
root@csc3150:~/source/program1# ./program1 ./illegal_instr
Process start to fork
I'm the Parent Process, my pid = 30631
I'm the Child Process, my pid = 30632
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGILL program

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

```
root@csc3150:~/source/program1# ./program1 ./interrupt
Process start to fork
I'm the Parent Process, my pid = 30688
I'm the Child Process, my pid = 30689
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
```

```
root@csc3150:~/source/program1# ./program1 ./pipe
Process start to fork
I'm the Parent Process, my pid = 30745
I'm the Child Process, my pid = 30746
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
```

```
root@csc3150:~/source/program1# ./program1 ./stop
Process start to fork
I'm the Parent Process, my pid = 30784
I'm the Child Process, my pid = 30785
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGSTOP program

Parent process receives SIGCHLD signal
CHILD PROCESS STOPPED
```

```
root@csc3150:~/source/program1# ./program1 ./terminate
Process start to fork
I'm the Parent Process, my pid = 30822
I'm the Child Process, my pid = 30823
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
```



Task 2:

Environment Set Up

First, we need to build the right environment. I enlarged my virtual-box to 5 CPUS and 10G storage. Then install the 5.10.5 kernel and recompile it using the following commands:

$ sudo -s

$ wget https://mirror.tuna.tsinghua.edu.cn/kernel/v5.x/linux-5.10.5.tar.xz

$ apt-get install libncurses-dev gawk flex bison openssl libssl-dev dkms libelf-dev libudev-dev libpci-dev libiberty-dev autoconf llvm dwarves bc

$ cp linux-5.10.5.tar.xz /home

$ cd home

$ sudo tar xvf linux-5.10.5.tar.xz /home

$ cd /boot

$ ll | grep .config

Then copy the file to /home/linux-5.10.5

$ make mrproper

$ make clean

$ make menuconfig

Save the config and exit

$make –j$(nproc)

$make modules_install

$make install

After reboot, the kernel has been installed.

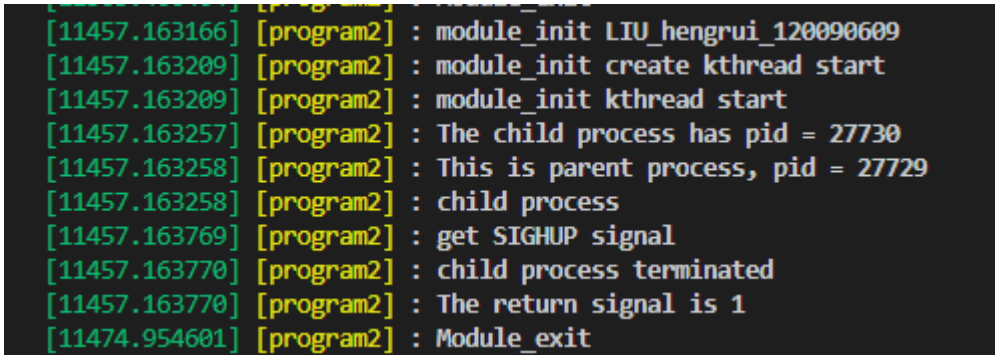
Then, I find the four functions in kernel files that we need to use in this assignment:

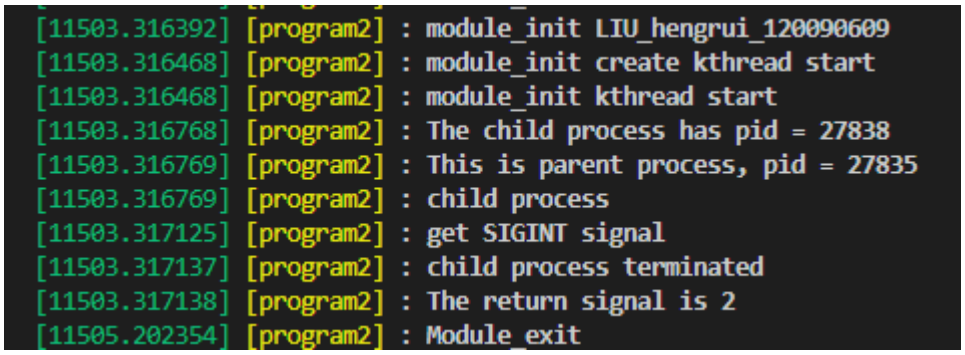
do_wait do_execve getname_kernel kernel_thread

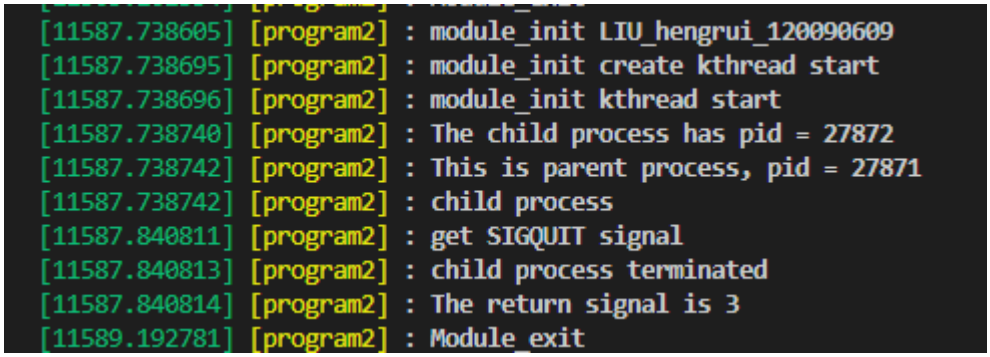
After EXPORT_SYMBOL(), and recompile the kernel, I can use them in my program2.c using extern.

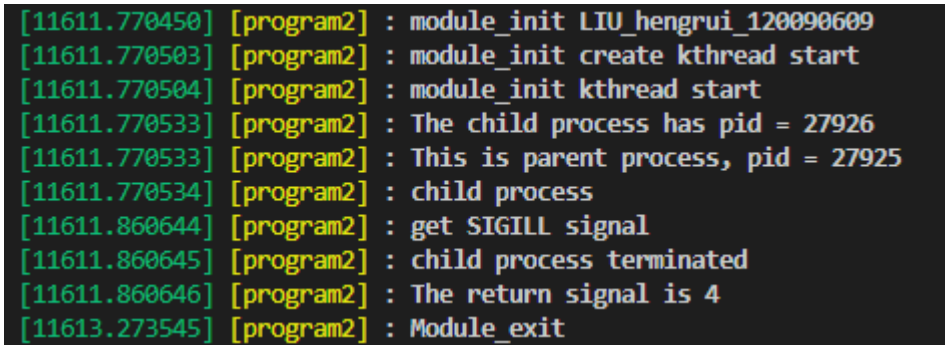
I designed my program2.c using my_wait(), my_exec(), my_fork(), program2_init() and program2_exit(). my_wait() is used to wait the execution of the child process and manage the signals raised by it. my_exec() is used to executed the child process. Here, germane_kernel using only 1 argument as the path. Do_execve also uses 1 arguments with others are NULL. The most import part is my_fork(), this function is used to fork a process, and wake my_execve and my_wait to execute and wait till it terminaties. The last part is program2_init() and program2_exit(), which is refer to the kernel codes of kernel_thread(). It uses the similar structure as task = kthread_create(), followed by if (!IS_ERR(task)) wake_up_process(task).

Here are my outputs (“Module_exit” is created by the command rmmod program2.ko, I think it’s meaningless to show it more than once, so I just ignore it in my screenshot.)









```
[11660.188833] [program2] : module_init LIU_hengrui_120090609
[11660.188900] [program2] : module_init create kthread start
[11660.188900] [program2] : module_init kthread start
[11660.188940] [program2] : The child process has pid = 28016
[11660.188950] [program2] : This is parent process, pid = 28015
[11660.188955] [program2] : child process
[11660.270186] [program2] : get SIGTRAP signal
[11660.270188] [program2] : child process terminated
[11660.270188] [program2] : The return signal is 5
[11661.483368] [program2] : Module_exit
```

```
[11685.103965] [program2] : module_init LIU_hengrui_120090609
[11685.104048] [program2] : module_init create kthread start
[11685.104049] [program2] : module_init kthread start
[11685.104093] [program2] : The child process has pid = 28066
[11685.104094] [program2] : This is parent process, pid = 28065
[11685.104095] [program2] : child process
[11685.186461] [program2] : get SIGABRT signal
[11685.186463] [program2] : child process terminated
[11685.186464] [program2] : The return signal is 6
[11686.456456] [program2] : Module_exit
```

```
[11711.484258] [program2] : module_init LIU_hengrui_120090609
[11711.484318] [program2] : module_init create kthread start
[11711.484318] [program2] : module_init kthread start
[11711.484344] [program2] : The child process has pid = 28133
[11711.484345] [program2] : This is parent process, pid = 28132
[11711.484345] [program2] : child process
[11711.566809] [program2] : get SIGBUS signal
[11711.566810] [program2] : child process terminated
[11711.566811] [program2] : The return signal is 7
[11712.745355] [program2] : Module_exit
```

```
[11736.784732] [program2] : module_init LIU_hengrui_120090609
[11736.784774] [program2] : module_init create kthread start
[11736.784774] [program2] : module_init kthread start
[11736.784805] [program2] : The child process has pid = 28184
[11736.784806] [program2] : This is parent process, pid = 28183
[11736.784806] [program2] : child process
[11736.866575] [program2] : get SIGFPE signal
[11736.866576] [program2] : child process terminated
[11736.866577] [program2] : The return signal is 8
[11737.937817] [program2] : Module_exit
```

```
[ 9854.824439] [program2] : module_init LIU_hengrui_120090609
[ 9854.824507] [program2] : module_init create kthread start
[ 9854.824507] [program2] : module_init kthread start
[ 9854.824637] [program2] : The child process has pid = 27478
[ 9854.824638] [program2] : This is parent process, pid = 27477
[ 9854.824639] [program2] : child process
[ 9854.825315] [program2] : get SIGKILL signal
[ 9854.825316] [program2] : child process terminated
[ 9854.825317] [program2] : The return signal is 9
[11363.400404] [program2] : Module_exit
```

```
[11758.499160] [program2] : module_init LIU_hengrui_120090609
[11758.499271] [program2] : module_init create kthread start
[11758.499271] [program2] : module_init kthread start
[11758.499372] [program2] : The child process has pid = 28234
[11758.499372] [program2] : This is parent process, pid = 28233
[11758.499373] [program2] : child process
[11758.499965] [program2] : get SIGUSR1 signal
[11758.499966] [program2] : child process terminated
[11758.499967] [program2] : The return signal is 10
[11760.262606] [program2] : Module_exit
```

```
[11794.009816] [program2] : module_init LIU_hengrui_120090609
[11794.009945] [program2] : module_init create kthread start
[11794.009945] [program2] : module_init kthread start
[11794.010055] [program2] : The child process has pid = 28282
[11794.010056] [program2] : This is parent process, pid = 28281
[11794.010056] [program2] : child process
[11794.092244] [program2] : get SIGSEGV signal
[11794.092246] [program2] : child process terminated
[11794.092246] [program2] : The return signal is 11
[11795.469908] [program2] : Module_exit
```

```
[11814.211350] [program2] : module_init LIU_hengrui_120090609
[11814.211417] [program2] : module_init create kthread start
[11814.211418] [program2] : module_init kthread start
[11814.211933] [program2] : The child process has pid = 28333
[11814.211935] [program2] : This is parent process, pid = 28331
[11814.211935] [program2] : child process
[11814.212267] [program2] : get SIGUSR2 signal
[11814.212268] [program2] : child process terminated
[11814.212268] [program2] : The return signal is 12
[11818.643924] [program2] : Module_exit
```

```
[11836.496378] [program2] : module_init LIU_hengrui_120090609
[11836.496512] [program2] : module_init create kthread start
[11836.496513] [program2] : module_init kthread start
[11836.496559] [program2] : The child process has pid = 28399
[11836.496559] [program2] : This is parent process, pid = 28398
[11836.496560] [program2] : child process
[11836.497056] [program2] : get SIGPIPE signal
[11836.497057] [program2] : child process terminated
[11836.497058] [program2] : The return signal is 13
[11837.958011] [program2] : Module_exit
```

```
[11859.884350] [program2] : module_init LIU_hengrui_120090609
[11859.884405] [program2] : module_init create kthread start
[11859.884406] [program2] : module_init kthread start
[11859.884698] [program2] : The child process has pid = 28469
[11859.884699] [program2] : This is parent process, pid = 28467
[11859.884700] [program2] : child process
[11859.884994] [program2] : get SIGALRM signal
[11859.884995] [program2] : child process terminated
[11859.884996] [program2] : The return signal is 14
[11861.359769] [program2] : Module_exit
```

```
[11879.606965] [program2] : module_init LIU_hengrui_120090609
[11879.607056] [program2] : module_init create kthread start
[11879.607056] [program2] : module_init kthread start
[11879.607090] [program2] : The child process has pid = 28516
[11879.607091] [program2] : This is parent process, pid = 28515
[11879.607091] [program2] : child process
[11879.607635] [program2] : get SIGTERM signal
[11879.607636] [program2] : child process terminated
[11879.607636] [program2] : The return signal is 15
[11880.791566] [program2] : Module_exit
```

```
[11926.247977] [program2] : module_init LIU_hengrui_120090609
[11926.249487] [program2] : module_init create kthread start
[11926.251310] [program2] : module_init kthread start
[11926.252823] [program2] : The child process has pid = 28602
[11926.254436] [program2] : This is parent process, pid = 28601
[11926.255969] [program2] : child process
[11931.253432] [program2] : normal termination
[11931.254875] [program2] : child process terminated
[11931.256506] [program2] : The return signal is 0
```

From this assignment, I learned how to compile the kernel, fork the child process, and write my code under kernel_mode instead of user_mode, as well as many Linux commands.