

CSC3150_A1_REPORT

Huang Lei_120090481

10/9/2021

Overview

This is the report of assignment 1 for CSC3150, 2022 fall semester. This project mainly focuses on learning and making use of processes and threads in two modes. One is user mode multi-process programming in program 1, the other is kernel mode multi-process programming in program 2. In **program 1**, we implement the functionality including forking a children process in user mode, executing the test program, while parent process is waiting for child process termination, and after child process terminating, parent process will handle and output different signals sent by children process. In **program 2**, we insert a modified model and create a kernel thread. Then fork a new process in kernel mode while parent process waiting until child process ends. Also, need to handle some signals. In **bonus**, this is multi-process programming problem. We are required to build a process tree and display all the process. It mainly uses recursive way to execute the file in terminal and also require us to communicate between each process.

Important Declaration

For Program 2, the path for test in my program is `"/tmp/test"`.

Environment

The environment of running my programs is the following:

OS version: Ubuntu 32-bit

```
vagrant@csc3150:~/csc3150/ASS2$ cat /etc/issue
Ubuntu 16.04.7 LTS \n \l
```

kernel version: Linux-5.10.146

```
root@csc3150:/home/vagrant/csc3150/ASS2# uname -r
5.10.146
```

gcc -version: gcc (Ubuntu 5.4.0-6ubuntu1~16.04.12) 5.4.0 20160609

```
linux-gnu --target=x86_64-linux-gnu
Thread model: posix
gcc version 5.4.0 20160609 (Ubuntu 5.4.0-6ubuntu1~16.04.12)
root@csc3150:/home/vagrant/csc3150/ASS2#
```

Program Execution Steps

Program 1

In order to execute the program 1, you should follow these steps :

```
$ cd /* directory where the program1.c located in */    (/home/vagrant/csc3150/ASS1/program1.c)
```

```
$ make
```

```
$ ./program1 ./test_program_name                    # you can only add one test here
```

Then you can see the output

```
$ make clean                                         # remember to clear
```

Program 2

First you should make sure you already install Ubuntu 5.10.146

Make sure you already install the modules, export and extern the function including do_wait, do_execve, getname_kernel, do_exit and kernel_thread.

Then you can do the following steps :

```
$ cd /* directory where your program2.c located in */  $ gcc -o test test.c
```

```
                # compile the test file
```

```
$ make                                                # generate the kernel file
```

```
$ sudo insmod program2.ko                            # insert the kernel module file into kernel
```

```
$ sudo rmmod program2                                # remove the kernel module file
```

```
$ dmesg | tail -n 10                                # print the last 10 message in the kernel log to
```

check you successfully finish the job

```
$ make clean                                         # remember to clear
```

Bonus

In order to execute the bonus, you should follow these steps :

```
$ cd /* directory where the pstree.c located in */
```

```
$ make
```

```
$ ./pstree -X                                         # X={c,p}, -c means disable compaction of  
identical sub trees, -p means show PIDs. Choose one argument and get the corresponding result.
```

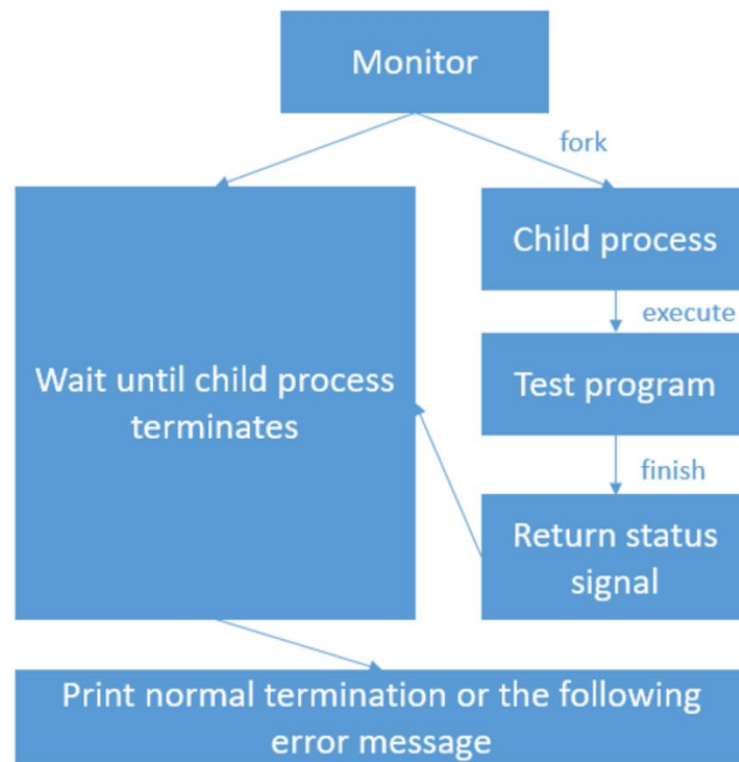
```
$ make clean                                         # remember to clear
```

Program Design

Here are the program designs. They offer basic ideas for each program and some important codes.

Program 1

The program 1 chart flow is:

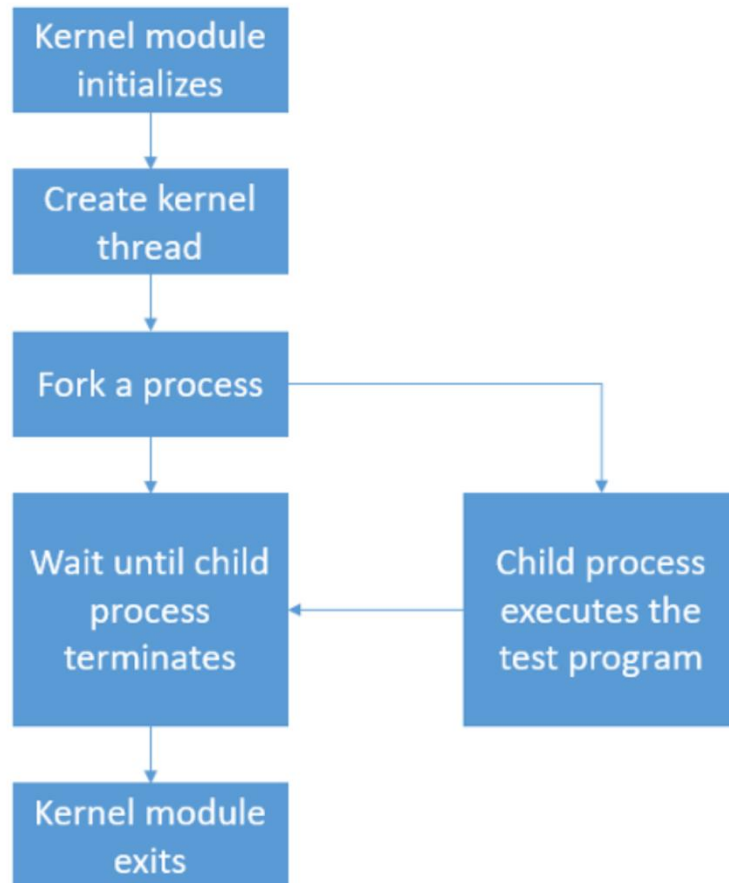


In this program, we first use `fork()` to create a process and its child process in user mode. Then use `execve()` to run the test file. The parent process will wait until the end of child process because of the function `waitpid()`, which can be replaced by function `wait()`. Finally, child process sends a signal to parent process telling which parts are wrong. We print this signal to see that.

In summary, 1. Fork a child process to execute test programs (15 of them) 2. Use `waitpid()` to let the parent process receives the `SIGCHLD` signal 3. Print out the termination information of child process (normal or abnormal)

program 2

The program 2 chart flow is:



For this problem, we need to fork the process in the kernel mode. First we made the kernel and kernel modules in the computer before we type make command. **Program2_init()** function is used to initialize the kernel and create the kernel thread. **my_fork()** is used to test the kernel. **my_fork()** is used to fork the process and get the pid for both parent and child process. Then enters **my_exec()**, which is used to open and run the test file. **my_wait()** function waits for the child process termination and send some signal to parent process. It does its job in the parent process. Within the **my_wait()** function, struct `wait_opts` is constructed and passed to `do_wait` as a parameter. Therefore, the parent process can check whether the child process is finished through the given child PID.

In summary, 1. Create a kernel thread and run `my_fork` function 2. Fork a process to execute `test.o` 3. Use `do_wait()` to let the parent process wait for the child process 4. Print out pid of both parent and child processes 5. Catch the signal raised by the child process and print out 6. Recompile the Linux kernel source code to use its functions

bonus

For this problem, this is a typical multi-process problem in user mode. The difficulty for this problem is how to create a process tree and how to print out the final information for those tree. For the first question, I use `bulid_tree()` and some other methods to implement it. It is a recursive function, it will

create a fork process tree. For the second one, I define a method called `print_tree()`, which prints node through linked list. At the beginning, the file will print the arguments input to see which mode it should print. Like if the argument is “-p”, then the tree will print pid of every node. Then, it uses struct DIR to open “/proc” file, which contains all the information of processes. Then we create node, including pid and its name. Use recursion to find thread in “/proc/#pid/task” file. Also create node for every one of them. After all nodes have been built, we create the tree node by node. Finally, print out information of every node in certain format, the similar one with the linux command “pstree”.

Development environment set up

Set up VM

- Follow the PPT of tutorial I, first install virtualbox and vagrant, then reboot the machine.
- Set up a directory for csc3150 (make sure the full path does not include space or Chinese, e.g. D:\csc3150).
- Launch powershell with Administrator privilege (run as administrator) and change current directory to the one you set up (e.g. `cd D:\csc3150`).
- Execute “`vagrant init cyzhu/csc3150`”
- Then execute “`vagrant up`”. It may take a while to download the system image. After that a virtualbox window may pop up. Leave it open but put it aside.

Set up VS Code

- After installing, go to the remote explorer tab, click config in SSH-TARGETS.
- Now, go back to powershell (make sure you are still in the csc3150 directory) and execute `vagrant ssh-config`. Copy everything but the last line (LogLevel) to the ssh config and save the file, as is demonstrated in the picture.
- Now you can find SSH Target called default (if not, may sure you have save the config or you can click the refresh button). Click the icon to connect to the VM and launch a new window.
- In the terminal you just opened, install essential dependencies and libraries: `sudo apt update && sudo apt install -y build-essential` (it may take a while). After it finishes, create a directory for the course: `mkdir -p ~/csc3150`. Then you can run your code there.

Compile kernel

- Download source code from “<http://www.kernel.org>”.
- Install Dependency and development tools using “`$sudo apt-get install libncurses-dev gawk flex bison openssl libssl-dev dkms libelf-dev libudev-dev libpci-dev libiberty-dev autoconf llvm dwarves`”.
- Create file in “/home” : “/home/seed/work”. Extract the source file to “/home/seed/work”
- Type following commands:

```
“$cp KERNEL_FILE.tar.xz /home/seed/work”
```

```
“$cd /home/seed/work”
```

```
“$sudo tar xvf KERNEL_FILE.tar.xz”
```

- Copy config from “/boot” to “/home/seed/work/KERNEL_FILE” (here mine is linux-5.10.146”
- Commands:
 - “\$sudo su”
 - “cd /home/seed/work/KERNEL_FILE”
 - “\$make mrproper”
 - “\$make clean”
 - “\$make menuconfig”
 - “\$make bzImage -j\$(nproc)”
 - “\$make modules -j\$(nproc)”
 - “\$make modules_install”
 - “\$make install”
- Reboot finally.

Screenshot of your program output

Program1:

```

● vagrant@csc3150:~/csc3150/ASS1$ make
cc -o program1 program1.c
● vagrant@csc3150:~/csc3150/ASS1$ ./program1 abort
Process start to fork
I'm the Parent Process, my pid = 3469
I'm the Child Process, my pid = 3470
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
Child process is abort by abort signal
● vagrant@csc3150:~/csc3150/ASS1$ ./program1 alarm
Process start to fork
I'm the Parent Process, my pid = 3494
I'm the Child Process, my pid = 3495
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
Child process is abort by alarm signal
● vagrant@csc3150:~/csc3150/ASS1$ ./program1 bus
Process start to fork
I'm the Parent Process, my pid = 3528
I'm the Child Process, my pid = 3529
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
Child process is abort by BUS signal

```

```

● vagrant@csc3150:~/csc3150/ASS1$ ./program1 floating
Process start to fork
I'm the Parent Process, my pid = 3544
I'm the Child Process, my pid = 3545
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
Child process is abort by SIGFPE signal
● vagrant@csc3150:~/csc3150/ASS1$ ./program1 hangup
Process start to fork
I'm the Parent Process, my pid = 3568
I'm the Child Process, my pid = 3569
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
Child process is hung up
● vagrant@csc3150:~/csc3150/ASS1$ ./program1 illegal_instr
Process start to fork
I'm the Parent Process, my pid = 3609
I'm the Child Process, my pid = 3610
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
Child process is abort by SIGILL signal

-----
● vagrant@csc3150:~/csc3150/ASS1$ ./program1 kill
Process start to fork
I'm the Parent Process, my pid = 3633
I'm the Child Process, my pid = 3634
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
Child process is abort by SIGKILL signal
● vagrant@csc3150:~/csc3150/ASS1$ ./program1 normal
Process start to fork
I'm the Parent Process, my pid = 3687
I'm the Child Process, my pid = 3688
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/ASS1$ ./program1 pipe
Process start to fork
I'm the Parent Process, my pid = 3773
I'm the Child Process, my pid = 3774
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
Child process is abort by SIGPIPE signal
-----

```


- **vagrant@csc3150:~/csc3150/ASS1\$./program1 quit**
 Process start to fork
 I'm the Parent Process, my pid = 3823
 I'm the Child Process, my pid = 3824
 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
 Child process is abort by SIGQUIT signal
- **vagrant@csc3150:~/csc3150/ASS1\$./program1 segment_fault**
 Process start to fork
 I'm the Parent Process, my pid = 3839
 I'm the Child Process, my pid = 3840
 Child process start to execute test program:
 -----CHILD PROCESS START-----
 This is the SIGSEGV program

 Parent process receives SIGCHLD signal
 Child process get SIGSEGV signal
 Child process is abort by SIGSEGV signal
- **vagrant@csc3150:~/csc3150/ASS1\$./program1 stop**
 Process start to fork
 I'm the Parent Process, my pid = 4031
 I'm the Child Process, my pid = 4032
 Child process start to execute test program:
 -----CHILD PROCESS START-----
 This is the SIGSTOP program

 Parent process receives SIGCHLD signal
 Child process get SIGSTOP signal
 Child process stopped
 CHILD EXECUTION STOPPED
- **vagrant@csc3150:~/csc3150/ASS1\$./program1 terminate**
 Process start to fork
 I'm the Parent Process, my pid = 4045
 I'm the Child Process, my pid = 4046
 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
 Child process is abort by SIGTERM signal
- **vagrant@csc3150:~/csc3150/ASS1\$./program1 trap**
 Process start to fork
 I'm the Parent Process, my pid = 4059
 I'm the Child Process, my pid = 4060
 Child process start to execute test program:
 -----CHILD PROCESS START-----
 This is the SIGTRAP program

 Parent process receives SIGCHLD signal
 Child process get SIGTRAP signal
 Child process is abort by SIGTRAP signal

- **vagrant@csc3150:~/csc3150/ASS1\$./program1** interrupt

```
Process start to fork
I'm the Parent Process, my pid = 4102
I'm the Child Process, my pid = 4103
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
Child process is abort by SIGINT_signal
```

Progeam2:

- **vagrant@csc3150:~/csc3150/ASS2\$ make**
make -C /lib/modules/5.10.146/build M=/home/vagrant/csc3150/ASS2 modules
make[1]: Entering directory '/home/seed/work/linux-5.10.146'
CC [M] /home/vagrant/csc3150/ASS2/program2.o
/home/vagrant/csc3150/ASS2/program2.c: In function 'my_wait':
/home/vagrant/csc3150/ASS2/program2.c:52:5: warning: ISO C90 forbids mixed declarations and code [-Wdeclaration-after-statement]
int a;
^
/home/vagrant/csc3150/ASS2/program2.c: In function 'my_fork':
/home/vagrant/csc3150/ASS2/program2.c:166:2: warning: ISO C90 forbids mixed declarations and code [-Wdeclaration-after-statement]
pid_t pid;
^
/home/vagrant/csc3150/ASS2/program2.c: In function 'my_wait':
/home/vagrant/csc3150/ASS2/program2.c:49:16: warning: 'status' is used uninitialized in this function [-Wuninitialized]
wo.wo_stat = status;
^
MODPOST /home/vagrant/csc3150/ASS2/Module.symvers
CC [M] /home/vagrant/csc3150/ASS2/program2.mod.o
LD [M] /home/vagrant/csc3150/ASS2/program2.ko
make[1]: Leaving directory '/home/seed/work/linux-5.10.146'
● **vagrant@csc3150:~/csc3150/ASS2\$ sudo insmod program2.ko**
● **vagrant@csc3150:~/csc3150/ASS2\$ sudo rmmod program2**
● **vagrant@csc3150:~/csc3150/ASS2\$ dmesg | tail -n 10**
[929.758403] [program2] : module_init {Huang Lei} {120090481}
[929.758404] [program2] : module_init create kthread starts
[929.758802] [program2] : module_init kthread starts
[929.759077] [program2] : The child process has pid =4970
[929.759078] [program2] : This is the parent process, pid=4969
[929.759140] [program2] : child process
[931.760759] [program2] : get SIGALARM signal
[931.760760] [program2] : child process has alarm error
[931.760761] [program2] : The return signal is 14
[935.054157] [program2] : Module_exit
○ **vagrant@csc3150:~/csc3150/ASS2\$**

- **vagrant@csc3150:~/csc3150/ASS2\$ make**
make -C /lib/modules/5.10.146/build M=/home/vagrant/csc3150/ASS2 modules
make[1]: Entering directory '/home/seed/work/linux-5.10.146'
make[1]: Leaving directory '/home/seed/work/linux-5.10.146'
● **vagrant@csc3150:~/csc3150/ASS2\$ sudo insmod program2.ko**
● **vagrant@csc3150:~/csc3150/ASS2\$ sudo rmmod program2**
● **vagrant@csc3150:~/csc3150/ASS2\$ dmesg | tail -n 10**
[1064.029213] [program2] : module_init {Huang Lei} {120090481}
[1064.029214] [program2] : module_init create kthread starts
[1064.029566] [program2] : module_init kthread starts
[1064.029830] [program2] : The child process has pid =5457
[1064.029830] [program2] : This is the parent process, pid=5456
[1064.030159] [program2] : child process
[1064.122238] [program2] : get SIGABRT signal
[1064.122239] [program2] : child process has abort error
[1064.122240] [program2] : The return signal is 6
[1067.084488] [program2] : Module_exit
○ **vagrant@csc3150:~/csc3150/ASS2\$**
-

```

● vagrant@csc3150:~/csc3150/ASS2$ make
make -C /lib/modules/5.10.146/build M=/home/vagrant/csc3150/ASS2 modules
make[1]: Entering directory '/home/seed/work/linux-5.10.146'
^[[A^[[Amake[1]: Leaving directory '/home/seed/work/linux-5.10.146'
● vagrant@csc3150:~/csc3150/ASS2$ sudo insmod program2.ko
● vagrant@csc3150:~/csc3150/ASS2$ sudo rmmod program2
● vagrant@csc3150:~/csc3150/ASS2$ dmesg | tail -n 10
[ 1480.697171] [program2] : module_init {Huang Lei} {120090481}
[ 1480.697173] [program2] : module_init create kthread starts
[ 1480.697511] [program2] : module_init kthread starts
[ 1480.697854] [program2] : The child process has pid =7396
[ 1480.697855] [program2] : This is the parent process, pid=7395
[ 1480.698250] [program2] : child process
[ 1480.698833] [program2] : get SIGKILL signal
[ 1480.698833] [program2] : child process is killed
[ 1480.698834] [program2] : The return signal is 9
[ 1482.183109] [program2] : Module_exit
○ vagrant@csc3150:~/csc3150/ASS2$ █

```

```

root@csc3150:/home/vagrant/csc3150/ASS2# make
make -C /lib/modules/5.10.146/build M=/home/vagrant/csc3150/ASS2 modules
make[1]: Entering directory '/home/seed/work/linux-5.10.146'
make[1]: Leaving directory '/home/seed/work/linux-5.10.146'
root@csc3150:/home/vagrant/csc3150/ASS2# sudo insmod program2.ko
root@csc3150:/home/vagrant/csc3150/ASS2# sudo rmmod program2
root@csc3150:/home/vagrant/csc3150/ASS2# dmesg | tail -n 15
[10726.599864] [program2] : child process
[10726.614554] [program2] : This is the parent process, pid=22852
[10726.646292] [program2] : get SIGSTOP signal
[10726.659283] [program2] : child process stopped
[10726.673123] [program2] : The return signal is 19
[10728.428670] [program2] : Module_exit
[10810.544788] [program2] : module_init {Huang Lei} {120090481}
[10810.563682] [program2] : module_init create kthread starts
[10810.583953] [program2] : module_init kthread starts
[10810.600649] [program2] : The child process has pid =23254
[10810.600863] [program2] : child process
[10810.617300] [program2] : This is the parent process, pid=23253
[10810.647502] [program2] : child process exit normally
[10810.664301] [program2] : The return signal is 0
[10812.968065] [program2] : Module_exit
root@csc3150:/home/vagrant/csc3150/ASS2# █

```

```

● vagrant@csc3150:~/csc3150/ASS2$ make
make -C /lib/modules/5.10.146/build M=/home/vagrant/csc3150/ASS2 modules
make[1]: Entering directory '/home/seed/work/linux-5.10.146'
^[[Amake[1]: Leaving directory '/home/seed/work/linux-5.10.146'
● vagrant@csc3150:~/csc3150/ASS2$ sudo insmod program2.ko
● vagrant@csc3150:~/csc3150/ASS2$ sudo rmmod program2
● vagrant@csc3150:~/csc3150/ASS2$ dmesg | tail -n 10
[ 1315.913692] [program2] : module_init {Huang Lei} {120090481}
[ 1315.913693] [program2] : module_init create kthread starts
[ 1315.914112] [program2] : module_init kthread starts
[ 1315.914310] [program2] : The child process has pid =6244
[ 1315.914311] [program2] : This is the parent process, pid=6243
[ 1315.914539] [program2] : child process
[ 1315.915194] [program2] : get SIGHUP signal
[ 1315.915195] [program2] : child process is hung up
[ 1315.915195] [program2] : The return signal is 1
[ 1318.858943] [program2] : Module_exit
○ vagrant@csc3150:~/csc3150/ASS2$ 

```

1

```

● vagrant@csc3150:~/csc3150/ASS2$ make
make -C /lib/modules/5.10.146/build M=/home/vagrant/csc3150/ASS2 modules
make[1]: Entering directory '/home/seed/work/linux-5.10.146'
^[[Amake[1]: Leaving directory '/home/seed/work/linux-5.10.146'
● vagrant@csc3150:~/csc3150/ASS2$ sudo insmod program2.ko
● vagrant@csc3150:~/csc3150/ASS2$ sudo rmmod program2
● vagrant@csc3150:~/csc3150/ASS2$ dmesg | tail -n 10
[ 1372.244997] [program2] : module_init {Huang Lei} {120090481}
[ 1372.244998] [program2] : module_init create kthread starts
[ 1372.245422] [program2] : module_init kthread starts
[ 1372.245615] [program2] : The child process has pid =6645
[ 1372.245616] [program2] : This is the parent process, pid=6644
[ 1372.245761] [program2] : child process
[ 1372.334279] [program2] : get SIGILL signal
[ 1372.334280] [program2] : child process has illegal instruction error
[ 1372.334281] [program2] : The return signal is 4
[ 1373.932915] [program2] : Module_exit
○ vagrant@csc3150:~/csc3150/ASS2$ 

```

```

● vagrant@csc3150:~/csc3150/ASS2$ make
make -C /lib/modules/5.10.146/build M=/home/vagrant/csc3150/ASS2 modules
make[1]: Entering directory '/home/seed/work/linux-5.10.146'
^[[Amake[1]: Leaving directory '/home/seed/work/linux-5.10.146'
● vagrant@csc3150:~/csc3150/ASS2$ sudo insmod program2.ko
● vagrant@csc3150:~/csc3150/ASS2$ sudo rmmod program2
● vagrant@csc3150:~/csc3150/ASS2$ dmesg | tail -n 10
[ 1428.251097] [program2] : module_init {Huang Lei} {120090481}
[ 1428.251098] [program2] : module_init create kthread starts
[ 1428.251335] [program2] : module_init kthread starts
[ 1428.251486] [program2] : The child process has pid =7021
[ 1428.251487] [program2] : This is the parent process, pid=7020
[ 1428.251740] [program2] : child process
[ 1428.256643] [program2] : get SIGINT signal
[ 1428.256645] [program2] : terminal interrupt
[ 1428.256645] [program2] : The return signal is 2
[ 1429.931966] [program2] : Module_exit
○ vagrant@csc3150:~/csc3150/ASS2$ 

```

```

● vagrant@csc3150:~/csc3150/ASS2$ make
make -C /lib/modules/5.10.146/build M=/home/vagrant/csc3150/ASS2 modules
make[1]: Entering directory '/home/seed/work/linux-5.10.146'
^[[A^[[Amake[1]: Leaving directory '/home/seed/work/linux-5.10.146'
● vagrant@csc3150:~/csc3150/ASS2$ sudo insmod program2.ko
● vagrant@csc3150:~/csc3150/ASS2$ sudo rmmod program2
● vagrant@csc3150:~/csc3150/ASS2$ dmesg | tail -n 10
[ 2569.067641] [program2] : module_init {Huang Lei} {120090481}
[ 2569.101281] [program2] : module_init create kthread starts
[ 2569.119146] [program2] : module_init kthread starts
[ 2569.134581] [program2] : The child process has pid =10576
[ 2569.135019] [program2] : child process
[ 2569.148396] [program2] : This is the parent process, pid=10575
[ 2569.268943] [program2] : get SIGFPE signal
[ 2569.279372] [program2] : child process terminated
[ 2569.291510] [program2] : The return signal is 8
[ 2571.442795] [program2] : Module_exit
○ vagrant@csc3150:~/csc3150/ASS2$ █

● vagrant@csc3150:~/csc3150/ASS2$ make
make -C /lib/modules/5.10.146/build M=/home/vagrant/csc3150/ASS2 modules
make[1]: Entering directory '/home/seed/work/linux-5.10.146'
^[[A^[[A CC [M] /home/vagrant/csc3150/ASS2/program2.o
^[[A^[[A/home/vagrant/csc3150/ASS2/program2.c: In function 'my_wait':
/home/vagrant/csc3150/ASS2/program2.c:52:5: warning: ISO C90 forbids mixed declarations and code [-Wdeclaration-after-statement]
    int a;
    ^
/home/vagrant/csc3150/ASS2/program2.c: In function 'my_fork':
/home/vagrant/csc3150/ASS2/program2.c:166:2: warning: ISO C90 forbids mixed declarations and code [-Wdeclaration-after-statement]
    pid_t pid;
    ^
/home/vagrant/csc3150/ASS2/program2.c: In function 'my_wait':
/home/vagrant/csc3150/ASS2/program2.c:49:16: warning: 'status' is used uninitialized in this function [-Wuninitialized]
    wo.wo_stat = status;
    ^
MODPOST /home/vagrant/csc3150/ASS2/Module.symvers
^[[A CC [M] /home/vagrant/csc3150/ASS2/program2.mod.o
^[[A LD [M] /home/vagrant/csc3150/ASS2/program2.ko
make[1]: Leaving directory '/home/seed/work/linux-5.10.146'
● vagrant@csc3150:~/csc3150/ASS2$ sudo insmod program2.ko
● vagrant@csc3150:~/csc3150/ASS2$ sudo rmmod program2
● vagrant@csc3150:~/csc3150/ASS2$ dmesg | tail -n 10
[ 2423.284154] [program2] : module_init {Huang Lei} {120090481}
[ 2423.312828] [program2] : module_init create kthread starts
[ 2423.330740] [program2] : module_init kthread starts
[ 2423.347010] [program2] : The child process has pid =10102
[ 2423.347402] [program2] : child process
[ 2423.361719] [program2] : This is the parent process, pid=10101
[ 2423.451568] [program2] : get SIGTRAP signal
[ 2423.462286] [program2] : child process has trap error
[ 2423.475355] [program2] : The return signal is 5
[ 2424.938457] [program2] : Module_exit
○ vagrant@csc3150:~/csc3150/ASS2$ █

● vagrant@csc3150:~/csc3150/ASS2$ make
make -C /lib/modules/5.10.146/build M=/home/vagrant/csc3150/ASS2 modules
make[1]: Entering directory '/home/seed/work/linux-5.10.146'
^[[A^[[A^[[Amake[1]: Leaving directory '/home/seed/work/linux-5.10.146'
● vagrant@csc3150:~/csc3150/ASS2$ sudo insmod program2.ko
● vagrant@csc3150:~/csc3150/ASS2$ sudo rmmod program2
● vagrant@csc3150:~/csc3150/ASS2$ dmesg | tail -n 10
[ 1792.028423] [program2] : module_init {Huang Lei} {120090481}
[ 1792.047621] [program2] : module_init create kthread starts
[ 1792.066558] [program2] : module_init kthread starts
[ 1792.084791] [program2] : The child process has pid =8940
[ 1792.085109] [program2] : child process
[ 1792.101925] [program2] : This is the parent process, pid=8939
[ 1792.203567] [program2] : get SIGSEGV signal
[ 1792.217342] [program2] : child process has segmentation fault error
[ 1792.237969] [program2] : The return signal is 11
[ 1793.596844] [program2] : Module_exit
○ vagrant@csc3150:~/csc3150/ASS2$ █

```



```

● vagrant@csc3150:~/csc3150/ASS2$ make
make -C /lib/modules/5.10.146/build M=/home/vagrant/csc3150/ASS2 modules
make[1]: Entering directory '/home/seed/work/linux-5.10.146'
^[[A^[[Amake[1]: Leaving directory '/home/seed/work/linux-5.10.146'
● vagrant@csc3150:~/csc3150/ASS2$ sudo insmod program2.ko
● vagrant@csc3150:~/csc3150/ASS2$ sudo rmmod program2
● vagrant@csc3150:~/csc3150/ASS2$ dmesg | tail -n 10
[ 1997.942633] [program2] : module_init {Huang Lei} {120090481}
[ 1997.958315] [program2] : module_init create kthread starts
[ 1997.974392] [program2] : module_init kthread starts
[ 1997.988349] [program2] : The child process has pid =9710
[ 1997.988654] [program2] : child process
[ 1998.005548] [program2] : This is the parent process, pid=9709
[ 1998.037361] [program2] : get SIGTERM signal
[ 1998.051211] [program2] : child process is terminated
[ 1998.067373] [program2] : The return signal is 15
[ 1999.795692] [program2] : Module_exit
○ vagrant@csc3150:~/csc3150/ASS2$ █

```

```

● vagrant@csc3150:~/csc3150/ASS2$ make
make -C /lib/modules/5.10.146/build M=/home/vagrant/csc3150/ASS2 modules
make[1]: Entering directory '/home/seed/work/linux-5.10.146'
^[[A^[[Amake[1]: Leaving directory '/home/seed/work/linux-5.10.146'
● vagrant@csc3150:~/csc3150/ASS2$ sudo insmod program2.ko
● vagrant@csc3150:~/csc3150/ASS2$ sudo rmmod program2
● vagrant@csc3150:~/csc3150/ASS2$ dmesg | tail -n 10
[ 1736.053056] [program2] : module_init {Huang Lei} {120090481}
[ 1736.078707] [program2] : module_init create kthread starts
[ 1736.097788] [program2] : module_init kthread starts
[ 1736.113751] [program2] : The child process has pid =8563
[ 1736.113987] [program2] : child process
[ 1736.128085] [program2] : This is the parent process, pid=8562
[ 1736.243963] [program2] : get SIGQUIT signal
[ 1736.255006] [program2] : terminal quit
[ 1736.265546] [program2] : The return signal is 3
[ 1737.407819] [program2] : Module_exit
○ vagrant@csc3150:~/csc3150/ASS2$ █

```

```

● vagrant@csc3150:~/csc3150/ASS2$ make
make -C /lib/modules/5.10.146/build M=/home/vagrant/csc3150/ASS2 modules
make[1]: Entering directory '/home/seed/work/linux-5.10.146'
^[[A^[[A^[[Amake[1]: Leaving directory '/home/seed/work/linux-5.10.146'
● vagrant@csc3150:~/csc3150/ASS2$ sudo insmod program2.ko
● vagrant@csc3150:~/csc3150/ASS2$ sudo rmmod program2
● vagrant@csc3150:~/csc3150/ASS2$ dmesg | tail -n 10
[ 1792.217342] [program2] : child process has segmentation fault error
[ 1792.237969] [program2] : The return signal is 11
[ 1793.596844] [program2] : Module_exit
[ 1892.634671] [program2] : module_init {Huang Lei} {120090481}
[ 1892.655493] [program2] : module_init create kthread starts
[ 1892.674789] [program2] : module_init kthread starts
[ 1892.691740] [program2] : The child process has pid =9357
[ 1892.692052] [program2] : child process
[ 1892.707016] [program2] : This is the parent process, pid=9356
[ 1894.974475] [program2] : Module_exit
○ vagrant@csc3150:~/csc3150/ASS2$ █

```



```

● vagrant@csc3150:~/csc3150/ASS2$ make
make -C /lib/modules/5.10.146/build M=/home/vagrant/csc3150/ASS2 modules
make[1]: Entering directory '/home/seed/work/linux-5.10.146'
^[[A^[[Amake[1]: Leaving directory '/home/seed/work/linux-5.10.146'
● vagrant@csc3150:~/csc3150/ASS2$ sudo insmod program2.ko
● vagrant@csc3150:~/csc3150/ASS2$ sudo rmmod program2
● vagrant@csc3150:~/csc3150/ASS2$ dmesg | tail -n 10
[ 1671.247615] [program2] : module_init {Huang Lei} {120090481}
[ 1671.266741] [program2] : module_init create kthread starts
[ 1671.284877] [program2] : module_init kthread starts
[ 1671.300093] [program2] : The child process has pid =8185
[ 1671.300962] [program2] : child process
[ 1671.317679] [program2] : This is the parent process, pid=8184
[ 1671.348549] [program2] : get SIGPIPE signal
[ 1671.364953] [program2] : child process has pipe error
[ 1671.382060] [program2] : The return signal is 13
[ 1673.015518] [program2] : Module_exit
○ vagrant@csc3150:~/csc3150/ASS2$ █

● vagrant@csc3150:~/csc3150/ASS2$ make
make -C /lib/modules/5.10.146/build M=/home/vagrant/csc3150/ASS2 modules
make[1]: Entering directory '/home/seed/work/linux-5.10.146'
^[[A^[[Amake[1]: Leaving directory '/home/seed/work/linux-5.10.146'
● vagrant@csc3150:~/csc3150/ASS2$ sudo insmod program2.ko
● vagrant@csc3150:~/csc3150/ASS2$ sudo rmmod program2
● vagrant@csc3150:~/csc3150/ASS2$ dmesg | tail -n 10
[ 2622.651265] [program2] : module_init {Huang Lei} {120090481}
[ 2622.668323] [program2] : module_init create kthread starts
[ 2622.682553] [program2] : module_init kthread starts
[ 2622.696236] [program2] : The child process has pid =10928
[ 2622.696401] [program2] : child process
[ 2622.712693] [program2] : This is the parent process, pid=10927
[ 2622.827874] [program2] : get SIGBUS signal
[ 2622.841547] [program2] : child process terminated
[ 2622.857879] [program2] : The return signal is 7
[ 2624.282143] [program2] : Module_exit
○ vagrant@csc3150:~/csc3150/ASS2$ █

```

```

make[1]: Leaving directory '/home/seed/work/linux-5.10.146'
root@csc3150:/home/vagrant/csc3150/ASS2# sudo insmod program2.ko
root@csc3150:/home/vagrant/csc3150/ASS2# sudo rmmod program2
root@csc3150:/home/vagrant/csc3150/ASS2# dmesg | tail -n 15
[ 9914.900311] status : 0, stat : 4991
[ 9914.912562] [program2] : get SIGSTOP signal
[ 9914.925784] [program2] : child process stopped
[ 9914.940598] [program2] : The return signal is 19
[ 9916.517412] [program2] : Module_exit
[10726.553779] [program2] : module_init {Huang Lei} {120090481}
[10726.570861] [program2] : module_init create kthread starts
[10726.586008] [program2] : module_init kthread starts
[10726.599536] [program2] : The child process has pid =22853
[10726.599864] [program2] : child process
[10726.614554] [program2] : This is the parent process, pid=22852
[10726.646292] [program2] : get SIGSTOP signal
[10726.659283] [program2] : child process stopped
[10726.673123] [program2] : The return signal is 19
[10728.428670] [program2] : Module_exit
root@csc3150:/home/vagrant/csc3150/ASS2# █

```

Bonus:

[illegible]

```

● vagrant@csc3150:~/csc3150/bonus$ ./pstree -c
argc:2
argv:./pstree argv:-c Have option: -c
systemd
├──systemd-journal
├──rsyslogd
├──lvmetad
├──systemd-udev
├──dhclient
├──iscsid
├──iscsid
├──systemd-logind
├──lxcfs
├──acpid
├──cron
├──dbus-daemon
├──accounts-daemon
├──atd
├──sshd
│   ├──sshd
│   │   ├──sshd
│   │   │   ├──bash
│   │   │   │   ├──sh
│   │   │   │   │   ├──node
│   │   │   │   │   │   ├──node
│   │   │   │   │   │   │   ├──bash
│   │   │   │   │   │   │   │   ├──pstree
│   │   │   │   │   │   │   │   ├──node
│   │   │   │   │   │   │   │   └──node
│   │   │   │   │   └──sleep
│   │   ├──sshd
│   │   │   ├──sshd
│   │   │   │   ├──bash
│   │   │   │   │   └──sleep
│   │   ├──sshd
│   │   │   ├──sshd
│   │   │   │   ├──bash
│   │   │   │   │   └──sleep
│   │   ├──sshd
│   │   │   ├──sshd
│   │   │   │   ├──bash
│   │   │   │   │   └──sleep
│   │   ├──sshd
│   │   │   ├──sshd
│   │   │   │   ├──bash
│   │   │   │   │   └──sleep
│   │   ├──sshd
│   │   │   ├──sshd
│   │   │   │   ├──bash
│   │   │   │   │   └──sleep
│   │   └──sshd
│   │       ├──sshd
│   │       │   ├──bash
│   │       │   │   └──sleep
│   └──mdadm
├──unattended-upgr
├──polkitd
├──irqbalance
├──agetty
├──agetty
├──systemd (sd-pam)
├──test
├──test
├──test
└──kthreadd
    └──rcu_gp

```

```

● vagrant@csc3150:~/csc3150/bonus$ ./pstree -p
argc:2
argv:./pstree argv:-p Have option: -p
(0)──systemd(1)
├──systemd-journal(421)
├──rsyslogd(1019)
├──lvmetad(431)
├──systemd-udev(449)
├──dhclient(872)
├──iscsid(1014)
├──iscsid(1015)
├──{in:imuxsock}(1022)
├──{in:imklog}(1023)
├──{rs:main}(1024)
├──systemd-logind(1020)
├──lxcfs(1025)
├──{lxcfs}(1058)
├──{lxcfs}(1059)
├──acpid(1026)
├──cron(1030)
├──dbus-daemon(1037)
├──accounts-daemon(1046)
├──{gmain}(1050)
├──{gdbus}(1052)
├──atd(1047)
├──sshd(1061)
│   ├──sshd(1472)
│   │   ├──sshd(1511)
│   │   │   ├──bash(1512)
│   │   │   │   ├──sh(1557)
│   │   │   │   │   ├──node(1567)
│   │   │   │   │   │   ├──node(1653)
│   │   │   │   │   │   │   ├──bash(24887)
│   │   │   │   │   │   │   │   ├──pstree(25084)
│   │   │   │   │   │   │   │   ├──node(24848)
│   │   │   │   │   │   │   │   ├──node(24842)
│   │   │   │   │   │   │   │   ├──{node}(1654)
│   │   │   │   │   │   │   │   ├──{node}(1655)
│   │   │   │   │   │   │   │   ├──{node}(1656)
│   │   │   │   │   │   │   │   ├──{node}(1657)
│   │   │   │   │   │   │   │   ├──{node}(1658)
│   │   │   │   │   │   │   │   ├──{node}(1662)
│   │   │   │   │   │   │   │   ├──{node}(1674)
│   │   │   │   │   │   │   │   ├──{node}(1675)
│   │   │   │   │   │   │   │   ├──{node}(1676)
│   │   │   │   │   │   │   │   ├──{node}(1677)
│   │   │   │   │   │   │   │   ├──{node}(24888)
│   │   │   │   │   │   │   │   ├──{node}(24843)
│   │   │   │   │   │   │   │   ├──{node}(24844)
│   │   │   │   │   │   │   │   ├──{node}(24845)
│   │   │   │   │   │   │   │   ├──{node}(24846)
│   │   │   │   │   │   │   │   ├──{node}(24847)
│   │   │   │   │   │   │   │   ├──{node}(24854)
│   │   │   │   │   │   │   │   ├──{node}(24856)

```

What I have learned in this project

The environment is the most difficult part i have encountered. I spent almost two whole days in setting the kernel environment (compile). At first, the default root file is in disk C. And when I install some files in the virtual machine, it already fills all the space. So I have no choice but delete the whole virtual machine and vagrant files. I download them all in disk D and set the default download root to disk D. Then it works well. Also, one day when I open the virtual machine, trying to connect the SSH host, it suddenly crashed. I have not figured it out until today. But the file I have already finished got lost. So I learn from this that I should always keep a copy of my work whenever I shut them off. This is really important.

What's more, googling is of great significance. I almost know nothing about kernel in the beginning. But after reading some information on the Internet, I work out lots of difficulties.