

Report for CSC3150 Project 1

Name: 曲恒毅, Hengyi Qu

Student id: 120090673

1. How did you design your program?

Task 1

I refer the chart flow to design my code.

Firstly, I fork the process and get the pid of each process. According to the pid's range. I identify the parent process and child process. And use waitpid command to implement that the child process should execute first, and the parent process can only be executed when the child process is done. During the process, I print some information to get their running order. Finally, I print the signal information or the terminated information of the child process.

Task 2

I firstly use the kernel thread to create a new thread. Then I use the my_fork() function to fork a process. In the my_fork(), I use my_kernel_clone() to clone the kernel. And use my_execve() to execute the test program in the child process. Then I use my_wait() to wait the child process finish. Then I exit the kernel module.

2. How to set up your development environment, including how to compile kernel?

I firstly use the mac M1 to config the environment. After I finish installing the UTM. I fail to install the kernel. So I change my work environment to another x86_64 Windows computer.

I firstly install Oracle VM VirtualBox. Install and set up vagrant, and vagrant ssh successfully in powershell. Then I set up the vscode to ssh the virtual machine conveniently with plugins. I set up the virtual machine's memory and the number of CPU cores (for my computer, 6G and 4 cores), to compile the kernel more quickly. In the virtual machine, I apt-get some dependent commands. And use make to config the kernel file, which is wget from the website. Then I compile it and wait about 1.5 hours. And install the kernel. Finally I reboot and load the new kernel. By the way, since program2 need some extern commands. I rebuild my kernel with adding some export_symbol in some kernel file.

3. Screenshot of your program output.

task1

abort

```
root@csc3150:/home/vagrant/source/program1# ./program1 ./abort
Process start to fork
I'm the Parent Process, my pid = 8411
I'm the Child Process, my pid = 8412
Child process start to execute test program:
-----CHILD PROCESS START-----
This is the SIGABRT program

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

alarm

```
root@csc3150:/home/vagrant/source/program1# ./program1 ./alarm
Process start to fork
I'm the Parent Process, my pid = 8531
I'm the Child Process, my pid = 8532
Child process start to execute test program:
-----CHILD PROCESS START-----
This is the SIGALRM program

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

bus

```
root@csc3150:/home/vagrant/source/program1# ./program1 ./bus
Process start to fork
I'm the Parent Process, my pid = 8653
I'm the Child Process, my pid = 8654
Child process start to execute test program:
-----CHILD PROCESS START-----
This is the SIGBUS program

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

floating

```
root@csc3150:/home/vagrant/source/program1# ./program1 ./floating
Process start to fork
I'm the Parent Process, my pid = 8698
I'm the Child Process, my pid = 8699
Child process start to execute test program:
-----CHILD PROCESS START-----
This is the SIGFPE program

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

hangup

```
root@csc3150:/home/vagrant/source/program1# ./program1 ./hangup
Process start to fork
I'm the Parent Process, my pid = 8788
I'm the Child Process, my pid = 8789
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
```

illegal_instr

```
root@csc3150:/home/vagrant/source/program1# ./program1 ./illegal_instr
Process start to fork
I'm the Parent Process, my pid = 8834
I'm the Child Process, my pid = 8835
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
```

interrupt

```
root@csc3150:/home/vagrant/source/program1# ./program1 ./interrupt
Process start to fork
I'm the Parent Process, my pid = 8901
I'm the Child Process, my pid = 8902
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
```

kill

```
root@csc3150:/home/vagrant/source/program1# ./program1 ./kill
Process start to fork
I'm the Parent Process, my pid = 8987
I'm the Child Process, my pid = 8988
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
```

normal

```
root@csc3150:/home/vagrant/source/program1# ./program1 ./normal
Process start to fork
I'm the Parent Process, my pid = 9010
I'm the Child Process, my pid = 9011
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
```

pipe

```
root@csc3150:/home/vagrant/source/program1# ./program1 ./pipe
Process start to fork
I'm the Parent Process, my pid = 9033
I'm the Child Process, my pid = 9034
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
```

quit

```
root@csc3150:/home/vagrant/source/program1# ./program1 ./quit
Process start to fork
I'm the Parent Process, my pid = 9077
I'm the Child Process, my pid = 9078
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
```

segment_fault

```
root@csc3150:/home/vagrant/source/program1# ./program1 ./segment_fault
Process start to to fork
I'm the Parent Process, my pid = 9101
I'm the Child Process, my pid = 9102
Child process start to execute test program:
-----CHILD PROCESS START-----
This is the SIGSEGV program

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

stop

```
root@csc3150:/home/vagrant/source/program1# ./program1 ./stop
Process start to fork
I'm the Parent Process, my pid = 9146
I'm the Child Process, my pid = 9147
Child process start to execute test program:
-----CHILD PROCESS START-----
This is the SIGSTOP program

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

terminate

```
root@csc3150:/home/vagrant/source/program1# ./program1 ./terminate
Process start to fork
I'm the Parent Process, my pid = 9190
I'm the Child Process, my pid = 9191
Child process start to execute test program:
-----CHILD PROCESS START-----
This is the SIGTERM program

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

trap

```
root@csc3150:/home/vagrant/source/program1# ./program1 ./trap
Process start to fork
I'm the Parent Process, my pid = 9204
I'm the Child Process, my pid = 9205
Child process start to execute test program:
-----CHILD PROCESS START-----
This is the SIGTRAP program

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

task2

bus

```
root@csc3150:/home/vagrant/source/program2# dmesg
[ 3797.559698] [program2] : Module_init
[ 3797.559701] [program2] : module_init {Hengyi Qu} {120090673}
[ 3797.559701] [program2] : module_init create kthread start
[ 3797.561287] [program2] : module_init kthread start
[ 3797.561634] [program2] : the child process has pid = 10691
[ 3797.561636] [program2] : This is the parent process, pid = 10690
[ 3797.561954] [program2] : child process
[ 3797.766505] [program2] : get SIGBUS signal
[ 3797.766506] [program2] : child process terminated
[ 3797.766507] [program2] : The return signal is 7
```

abort

```
root@csc3150:/home/vagrant/source/program2# dmesg
[ 4344.970186] [program2] : Module_exit
[ 4347.013406] [program2] : Module_init
[ 4347.013407] [program2] : module_init {Hengyi Qu} {120090673}
[ 4347.013408] [program2] : module_init create kthread start
[ 4347.013986] [program2] : module_init kthread start
[ 4347.014322] [program2] : the child process has pid = 11568
[ 4347.014323] [program2] : This is the parent process, pid = 11566
[ 4347.014385] [program2] : child process
[ 4347.214303] [program2] : get SIGABRT signal
[ 4347.214306] [program2] : child process terminated
[ 4347.214306] [program2] : The return signal is 6
```

alarm

```
root@csc3150:/home/vagrant/source/program2# dmesg
[ 4898.270174] [program2] : Module_exit
[ 4900.300895] [program2] : Module_init
[ 4900.300898] [program2] : module_init {Hengyi Qu} {120090673}
[ 4900.300898] [program2] : module_init create kthread start
[ 4900.302041] [program2] : module_init kthread start
[ 4900.302391] [program2] : the child process has pid = 13371
[ 4900.302394] [program2] : This is the parent process, pid = 13370
[ 4900.302772] [program2] : child process
[ 4902.305068] [program2] : get SIGALARM signal
[ 4902.305071] [program2] : child process terminated
[ 4902.305071] [program2] : The return signal is 14
```

floating


```
root@csc3150:/home/vagrant/source/program2# dmesg
[ 5025.545821] [program2] : Module_init
[ 5025.545824] [program2] : module_init {Hengyi Qu} {120090673}
[ 5025.545825] [program2] : module_init create kthread start
[ 5025.546873] [program2] : module_init kthread start
[ 5025.547066] [program2] : the child process has pid = 13493
[ 5025.547068] [program2] : This is the parent process, pid = 13492
[ 5025.547187] [program2] : child process
[ 5025.750491] [program2] : get SIGFPE signal
[ 5025.750493] [program2] : child process terminated
[ 5025.750494] [program2] : The return signal is 8
```

hangup

```
root@csc3150:/home/vagrant/source/program2# dmesg
[ 5120.509876] [program2] : Module_init
[ 5120.509878] [program2] : module_init {Hengyi Qu} {120090673}
[ 5120.509879] [program2] : module_init create kthread start
[ 5120.510501] [program2] : module_init kthread start
[ 5120.510712] [program2] : the child process has pid = 13553
[ 5120.510714] [program2] : This is the parent process, pid = 13552
[ 5120.511117] [program2] : child process
[ 5120.514351] [program2] : get SIGHUP signal
[ 5120.514354] [program2] : child process terminated
[ 5120.514396] [program2] : The return signal is 1
```

illegal_instr

```
root@csc3150:/home/vagrant/source/program2# dmesg
[ 5232.935200] [program2] : Module_init
[ 5232.935203] [program2] : module_init {Hengyi Qu} {120090673}
[ 5232.935203] [program2] : module_init create kthread start
[ 5232.935536] [program2] : module_init kthread start
[ 5232.935854] [program2] : the child process has pid = 13654
[ 5232.935856] [program2] : This is the parent process, pid = 13652
[ 5232.935860] [program2] : child process
[ 5233.141040] [program2] : get SIGILL signal
[ 5233.141042] [program2] : child process terminated
[ 5233.141042] [program2] : The return signal is 4
```

interrupt


```
root@csc3150:/home/vagrant/source/program2# dmesg
[ 5300.895319] [program2] : Module_init
[ 5300.895321] [program2] : module_init {Hengyi Qu} {120090673}
[ 5300.895322] [program2] : module_init create kthread start
[ 5300.895948] [program2] : module_init kthread start
[ 5300.896289] [program2] : the child process has pid = 13715
[ 5300.896290] [program2] : This is the parent process, pid = 13713
[ 5300.896293] [program2] : child process
[ 5300.897216] [program2] : get SIGINT signal
[ 5300.897218] [program2] : child process terminated
[ 5300.897219] [program2] : The return signal is 2
```

kill

```
root@csc3150:/home/vagrant/source/program2# dmesg
[ 5362.015592] [program2] : Module_init
[ 5362.015595] [program2] : module_init {Hengyi Qu} {120090673}
[ 5362.015597] [program2] : module_init create kthread start
[ 5362.017307] [program2] : module_init kthread start
[ 5362.017594] [program2] : the child process has pid = 13751
[ 5362.017595] [program2] : This is the parent process, pid = 13750
[ 5362.017601] [program2] : child process
[ 5362.019270] [program2] : get SIGKILL signal
[ 5362.019273] [program2] : child process terminated
[ 5362.019274] [program2] : The return signal is 9
```

normal

```
root@csc3150:/home/vagrant/source/program2# dmesg
[ 5748.387104] [program2] : Module_init
[ 5748.387107] [program2] : module_init {Hengyi Qu} {120090673}
[ 5748.387108] [program2] : module_init create kthread start
[ 5748.387698] [program2] : module_init kthread start
[ 5748.388504] [program2] : the child process has pid = 13833
[ 5748.388506] [program2] : This is the parent process, pid = 13832
[ 5748.388510] [program2] : child process
[ 5748.389696] [program2] : child process exit normally
[ 5748.389697] [program2] : The return signal is 0
```

pipe

```
root@csc3150:/home/vagrant/source/program2# dmesg
[ 5844.389024] [program2] : Module_init
[ 5844.389027] [program2] : module_init {Hengyi Qu} {120090673}
[ 5844.389028] [program2] : module_init create kthread start
[ 5844.390086] [program2] : module_init kthread start
[ 5844.390347] [program2] : the child process has pid = 13975
[ 5844.390349] [program2] : This is the parent process, pid = 13973
[ 5844.390435] [program2] : child process
[ 5844.392650] [program2] : get SIGPIPE signal
[ 5844.392652] [program2] : child process terminated
[ 5844.392653] [program2] : The return signal is 13
```

quit

```
root@csc3150:/home/vagrant/source/program2# dmesg
[ 5909.923375] [program2] : Module_init
[ 5909.923378] [program2] : module_init {Hengyi Qu} {120090673}
[ 5909.923380] [program2] : module_init create kthread start
[ 5909.924849] [program2] : module_init kthread start
[ 5909.925268] [program2] : the child process has pid = 14063
[ 5909.925271] [program2] : This is the parent process, pid = 14061
[ 5909.925534] [program2] : child process
[ 5910.120198] [program2] : get SIGQUIT signal
[ 5910.120200] [program2] : child process terminated
[ 5910.120201] [program2] : The return signal is 3
```

segment_fault

```
root@csc3150:/home/vagrant/source/program2# dmesg
[ 5953.534260] [program2] : Module_init
[ 5953.534263] [program2] : module_init {Hengyi Qu} {120090673}
[ 5953.534264] [program2] : module_init create kthread start
[ 5953.535115] [program2] : module_init kthread start
[ 5953.535492] [program2] : the child process has pid = 14142
[ 5953.535494] [program2] : This is the parent process, pid = 14140
[ 5953.535610] [program2] : child process
[ 5953.731348] [program2] : get SIGSEGV signal
[ 5953.731350] [program2] : child process terminated
[ 5953.731351] [program2] : The return signal is 11
```

terminate

```
root@csc3150:/home/vagrant/source/program2# dmesg
[ 6058.442725] [program2] : Module_exit
[ 6060.207213] [program2] : Module_init
[ 6060.207216] [program2] : module_init {Hengyi Qu} {120090673}
[ 6060.207217] [program2] : module_init create kthread start
[ 6060.207828] [program2] : module_init kthread start
[ 6060.208568] [program2] : the child process has pid = 14366
[ 6060.208570] [program2] : This is the parent process, pid = 14364
[ 6060.208912] [program2] : child process
[ 6060.210586] [program2] : get SIGTERM signal
[ 6060.210588] [program2] : child process terminated
[ 6060.210589] [program2] : The return signal is 15
```

trap

```
root@csc3150:/home/vagrant/source/program2# dmesg
[ 6123.635697] [program2] : Module_exit
[ 6124.935471] [program2] : Module_init
[ 6124.935474] [program2] : module_init {Hengyi Qu} {120090673}
[ 6124.935474] [program2] : module_init create kthread start
[ 6124.936551] [program2] : module_init kthread start
[ 6124.936795] [program2] : the child process has pid = 14445
[ 6124.936796] [program2] : This is the parent process, pid = 14444
[ 6124.936919] [program2] : child process
[ 6125.139921] [program2] : get SIGTRAP signal
[ 6125.139923] [program2] : child process terminated
[ 6125.139924] [program2] : The return signal is 5
```

stop

```
root@csc3150:/home/vagrant/source/program2# dmesg
[ 6308.975755] [program2] : Module_exit
[ 6310.641905] [program2] : Module_init
[ 6310.641908] [program2] : module_init {Hengyi Qu} {120090673}
[ 6310.641909] [program2] : module_init create kthread start
[ 6310.642860] [program2] : module_init kthread start
[ 6310.643297] [program2] : the child process has pid = 15552
[ 6310.643299] [program2] : This is the parent process, pid = 15551
[ 6310.643496] [program2] : child process
[ 6310.646568] [program2] : get SIGSTOP signal
[ 6310.646571] [program2] : child process terminated
[ 6310.646572] [program2] : The return signal is 19
```

4. What did you learn from the tasks?

1. I learn that how to find the usage of function by reading the source code. In this project, I read the kernel source code in elixir.bootlin.com for a long time to get the usage of kernel functions.

2. I get deeper understandings about how the kernel works, and the fork, waitpid, execve operation.
3. I enhance my coding ability during the project.