Assignment1 Report

Name:余永琦 ID:120090761

How did I design my program?

Basically, the two flow charts in the assignment description guide me to finish the program.

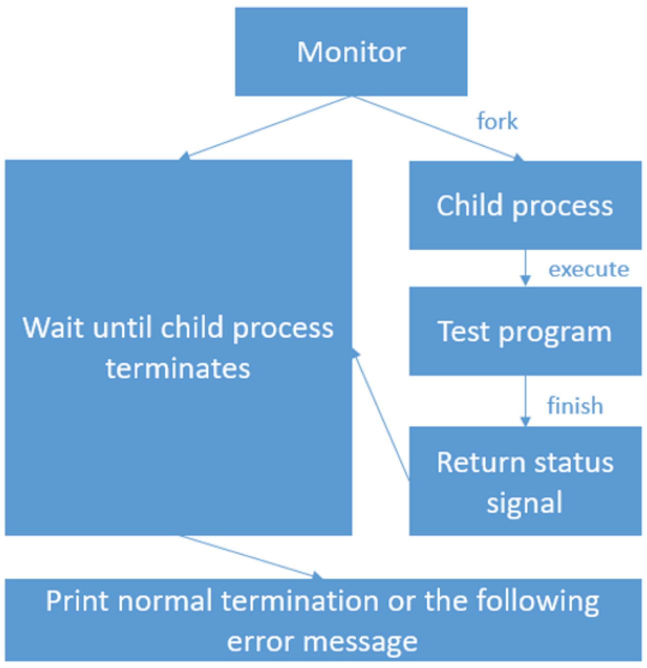

Monitor
fork
Child process
execute
Test program
finish
Return status signal
Wait until child process terminates
Print normal termination or the following error message

Chart1

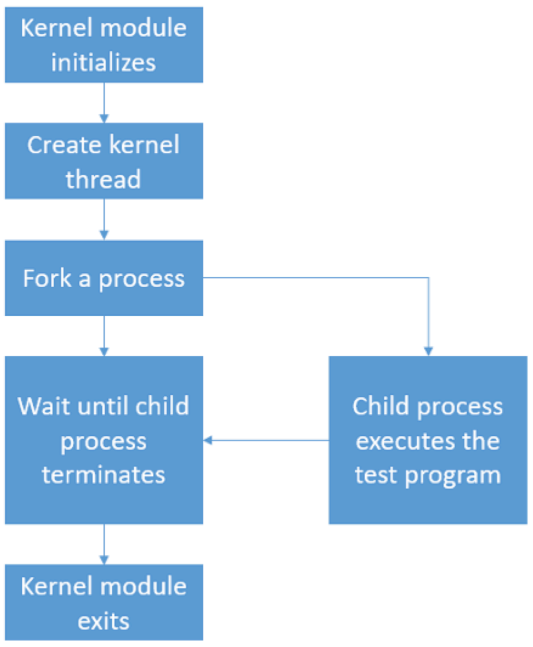

Kernel module initializes
Create kernel thread
Fork a process
Wait until child process terminates
Child process executes the test program
Kernel module exits

Chart2

Task1

In this task, we are asked to fork a child process and execute a program in the child process, and then return signal to the parent process. As Chart1 shows, we first need to fork the child process using the function fork(). The function fork() returns us a pid. If the return pid is -1, then there exists some errors. Else, with the value of this pid, we can distinguish the child process and parent process (the one with pid=0 is child process otherwise parent process). Now we have two processes.

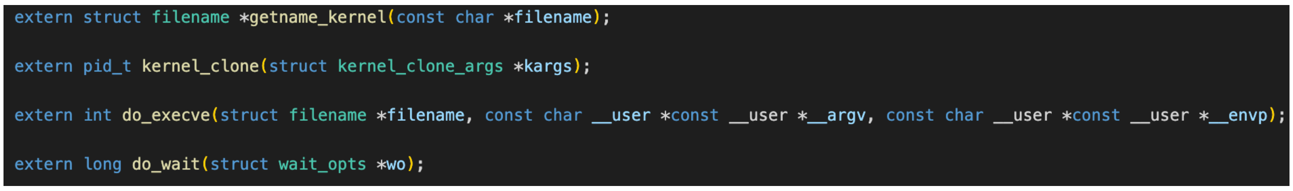
For the child process, we need to execute a test program and then return a signal. We can use the function "execve" to help us finish the job. We just put the filename of the program we want to run and all the arguments in it, set environment parameter to NULL, and then this function will do the job, it will execute the program and return the signal.

For the parent process, we need to wait until the child process terminates. The "waitpid" function can do this job. We pass the child process pid, a pointer to the status that record the signal as well as the option "WUNTRACED" to handle the stopped and terminated cases. After that, the variable status contains the information of return signal. Then we use "WIFEXITED", "WIFSIGNALED" and "WIFSTOPPED" functions to evaluate child process's status, and use "WEXITSTATUS", "WTERMSIG", "WSTOPSIG" to get the exact value.

Now we have finished Task1!

Task2

From Chart2 we can see that Task 2 asks us to do almost the same thing in Task1, however we need to do it by kernel module instead of the user aspect. Since we need to do it in kernel module, then we need to extern some function from the kernel to help. To extern them, we need to export them first(see the detail in next section).



```
extern struct filename *getname_kernel(const char *filename);

extern pid_t kernel_clone(struct kernel_clone_args *kargs);

extern int do_execve(struct filename *filename, const char __user *const __user *__argv, const char __user *const __user *__envp);

extern long do_wait(struct wait_opts *wo);
```

Chart3

In addition, I defined some functions my_wait() and my_execute() to do the job of wait() and execute() in Task1, also my_fork() similar to fork(). Now we get the most important functions my_fork(), my_wait() and my_execute() function, then using the **same logic in Task1** to handle child process and parent process can finish this Task. But we need to notice that instead of using function like “WIFEXITED” to handle the signal, we directly send the status to a map function to handle the returned signal.

In both tasks, there is a map function to handle the returned signal and print its corresponding information. **For the implementation details, please check my codes and see the comments.**

Bonus

In bonus, we are asked to implement the “pstree” command. Basically, we need to traverse the file in the /proc file system and print the tree out. In this task, we utilize the file /proc/PID/status to get many useful information, such a process’s pid, its parent id, its threads number and so on. By those information, we can build a tree by defining a data structure to store processes and creating links between them. Finally, use a recursive function to print the tree out.

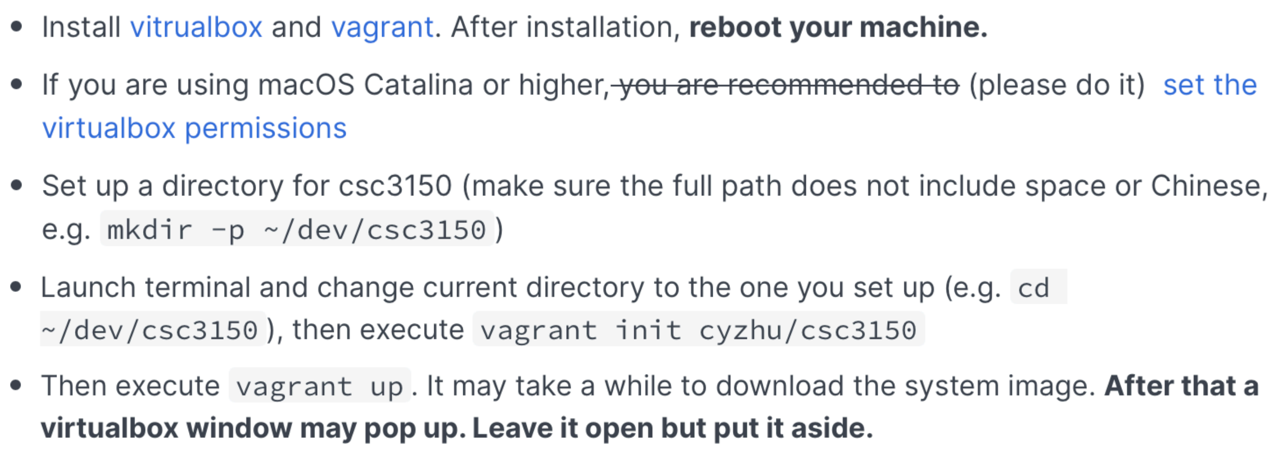
The options: “-p”, “-g”, “-c”, “-A”, “-n”, “-l”, “-V” are supported.

Some explanation: I did not compress the same nodes together, so the option -c is naturally finished. Also, I use “+”,“-”, “|”, “`” to print the tree, so the option -A is naturally finished. Besides, my program will automatically sort the process by pid, so the option -n is naturally finished, and “-l” is naturally finished as well.

How to set up my development environment, including how to compile kernel?

Set up environment

All the programs are run in the virtual machine “csc3150”. Tutorial shows how to set up the virtual machine:



- Install vitrualbox and vagrant. After installation, **reboot your machine.**
- If you are using macOS Catalina or higher, ~~you are recommended to~~ (please do it) set the virtualbox permissions
- Set up a directory for csc3150 (make sure the full path does not include space or Chinese, e.g. `mkdir -p ~/dev/csc3150`)
- Launch terminal and change current directory to the one you set up (e.g. `cd ~/dev/csc3150`), then 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.**

Follow those steps and we can set up the virtual environment. Each time we enter the file and type “vagrant up” to start the machine, and “vagrant halt” to stop it.

Version: Operating System: Ubuntu 16.04.7 LTS Kernel: Linux 5.10.146 gcc: 5.4.0

Compile kernel

First we need to download the 5.10.x version to update our kernel to meet the version requirement, by the command “wget https://mirror.tuna.tsinghua.edu.cn/kernel/v5.x/linux-5.10.145.tar.xz". Then, we need to decompress the file by typing “sudo tar xvf KERNEL_FILE.tar.xz” in terminal. Then we obtain a linux-5.10.146 folder.

Then we need to change some source code of the linux kernel because we need to call some functions in the kernel that are not open to the user mode in Task2, which means that our .c program cannot use them without exporting them first. Then we get into the folder and change the source code. This is also very simple, we just find the files that define the four functions we need to extern(the four functions are shown in Chart3) and add a “EXPORT_SYMBOL(function name)” below the function code. Remember to delete the “static” before the functions “do_wait” and “do_execute”, because exporting a static function is meaningless.

After changing the source code, we can now compile the kernel. We just enter the linux-5.10.146 directory and do the following steps:

First clean previous setting and start configuration, put “make mrproper”, “make clean” and “make menuconfig” in order to the terminal. In the last command we need to save the config and exit. Before we start, we need to copy the .config file of the old kernel to the new one.

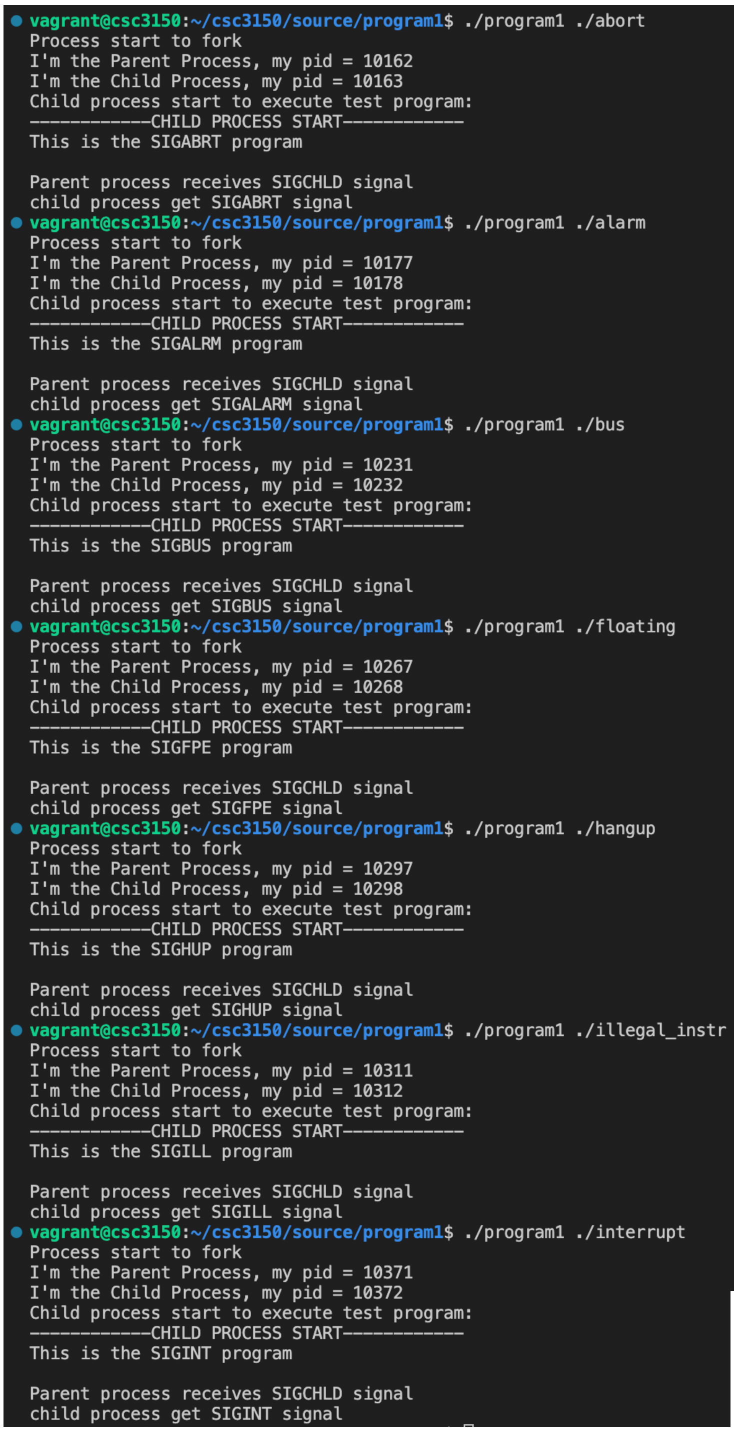
Then we build kernel image and modules by “make bzImage -j$(nproc)”, and then “make modules -j$(nproc)”. We can also replace the two commands in one “make –j$(nproc)”.

Finally install kernel modules “make modules_install” , install kernel “make install”. Then type “reboot” and choose the updated kernel.

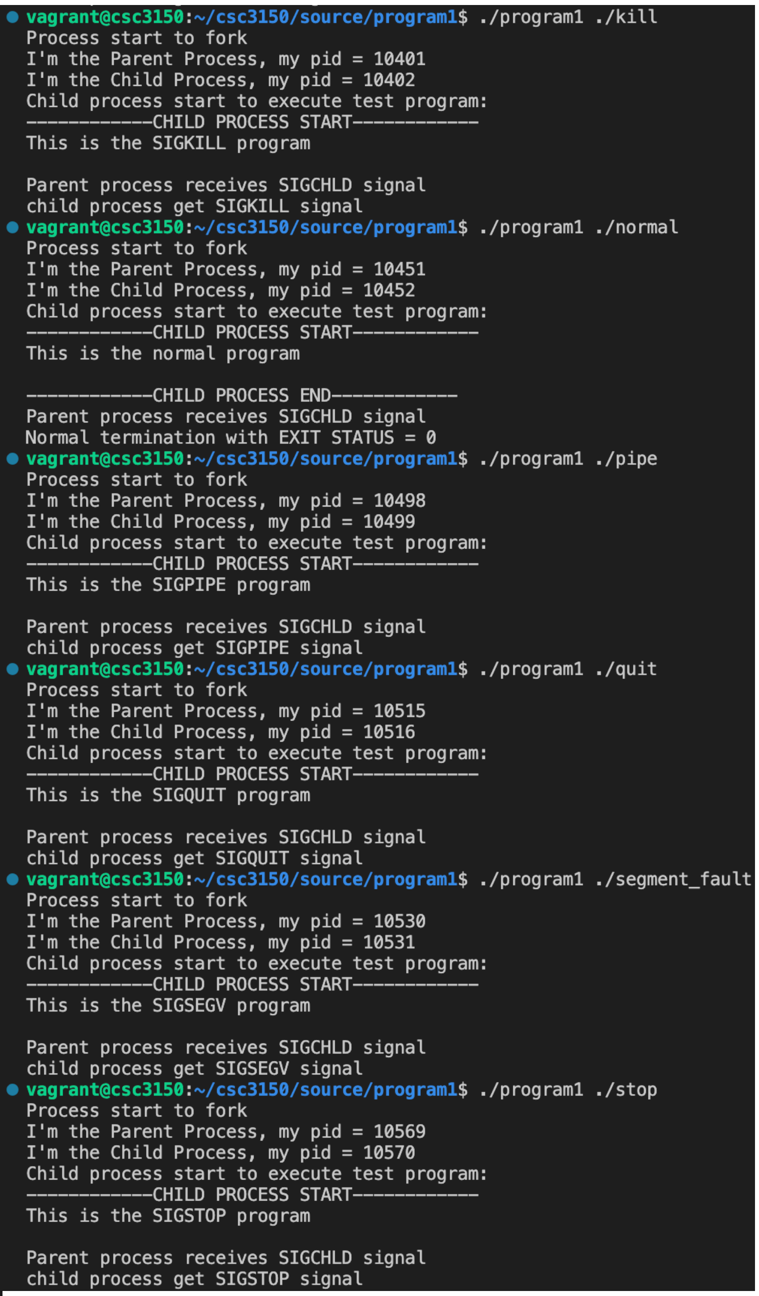
Now we finish setting the environment and compiling kernel.

Screen Shot

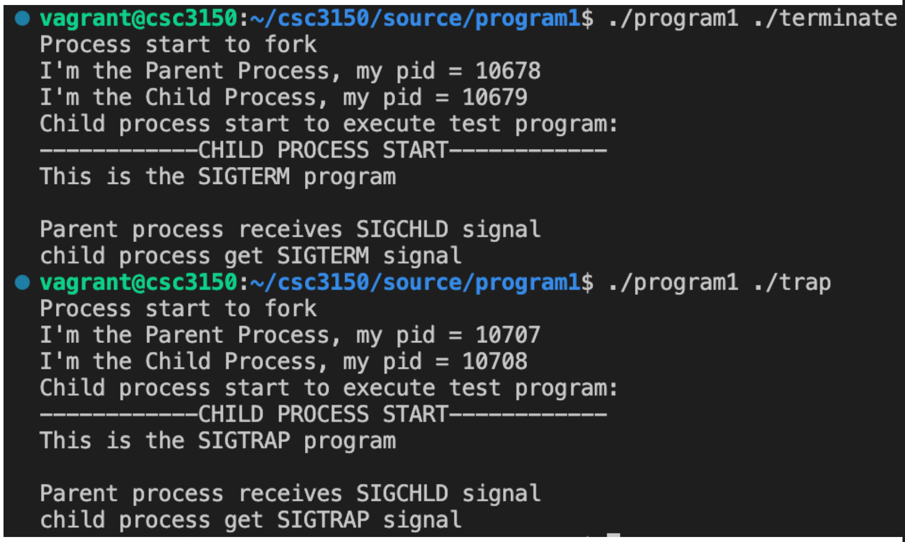
Task1



```
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./abort
Process start to fork
I'm the Parent Process, my pid = 10162
I'm the Child Process, my pid = 10163
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
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./alarm
Process start to fork
I'm the Parent Process, my pid = 10177
I'm the Child Process, my pid = 10178
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGALRM program

Parent process receives SIGCHLD signal
child process get SIGALARM signal
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./bus
Process start to fork
I'm the Parent Process, my pid = 10231
I'm the Child Process, my pid = 10232
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
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./floating
Process start to fork
I'm the Parent Process, my pid = 10267
I'm the Child Process, my pid = 10268
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
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./hangup
Process start to fork
I'm the Parent Process, my pid = 10297
I'm the Child Process, my pid = 10298
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
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./illegal_instr
Process start to fork
I'm the Parent Process, my pid = 10311
I'm the Child Process, my pid = 10312
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
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./interrupt
Process start to fork
I'm the Parent Process, my pid = 10371
I'm the Child Process, my pid = 10372
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
```

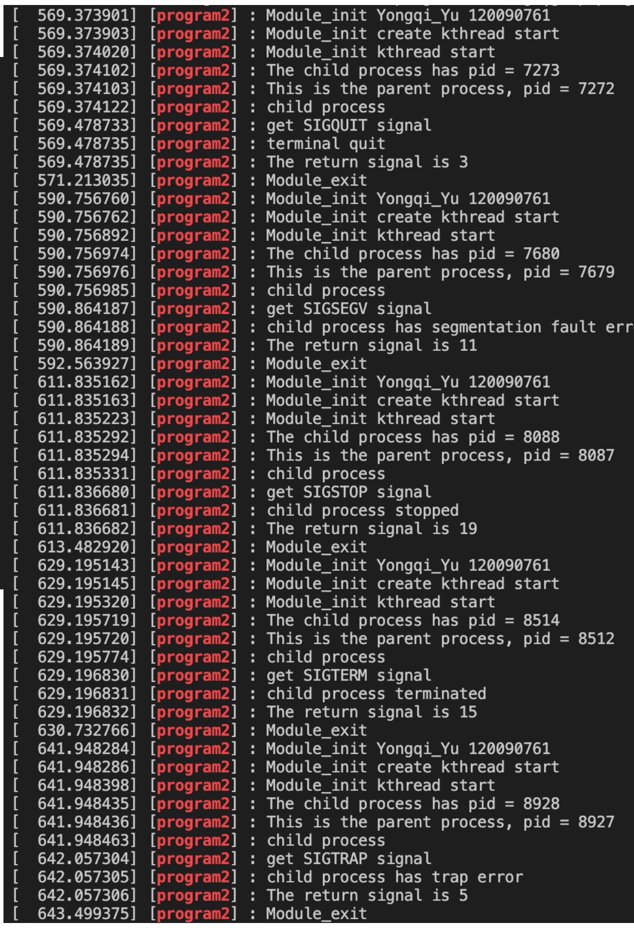


```
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./kill
Process start to fork
I'm the Parent Process, my pid = 10401
I'm the Child Process, my pid = 10402
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGKILL program

Parent process receives SIGCHLD signal
child process get SIGKILL signal
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./normal
Process start to fork
I'm the Parent Process, my pid = 10451
I'm the Child Process, my pid = 10452
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/source/program1$ ./program1 ./pipe
Process start to fork
I'm the Parent Process, my pid = 10498
I'm the Child Process, my pid = 10499
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
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./quit
Process start to fork
I'm the Parent Process, my pid = 10515
I'm the Child Process, my pid = 10516
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
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./segment_fault
Process start to fork
I'm the Parent Process, my pid = 10530
I'm the Child Process, my pid = 10531
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
vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./stop
Process start to fork
I'm the Parent Process, my pid = 10569
I'm the Child Process, my pid = 10570
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
```



```
● vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./terminate
  Process start to fork
  I'm the Parent Process, my pid = 10678
  I'm the Child Process, my pid = 10679
  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
● vagrant@csc3150:~/csc3150/source/program1$ ./program1 ./trap
  Process start to fork
  I'm the Parent Process, my pid = 10707
  I'm the Child Process, my pid = 10708
  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
```

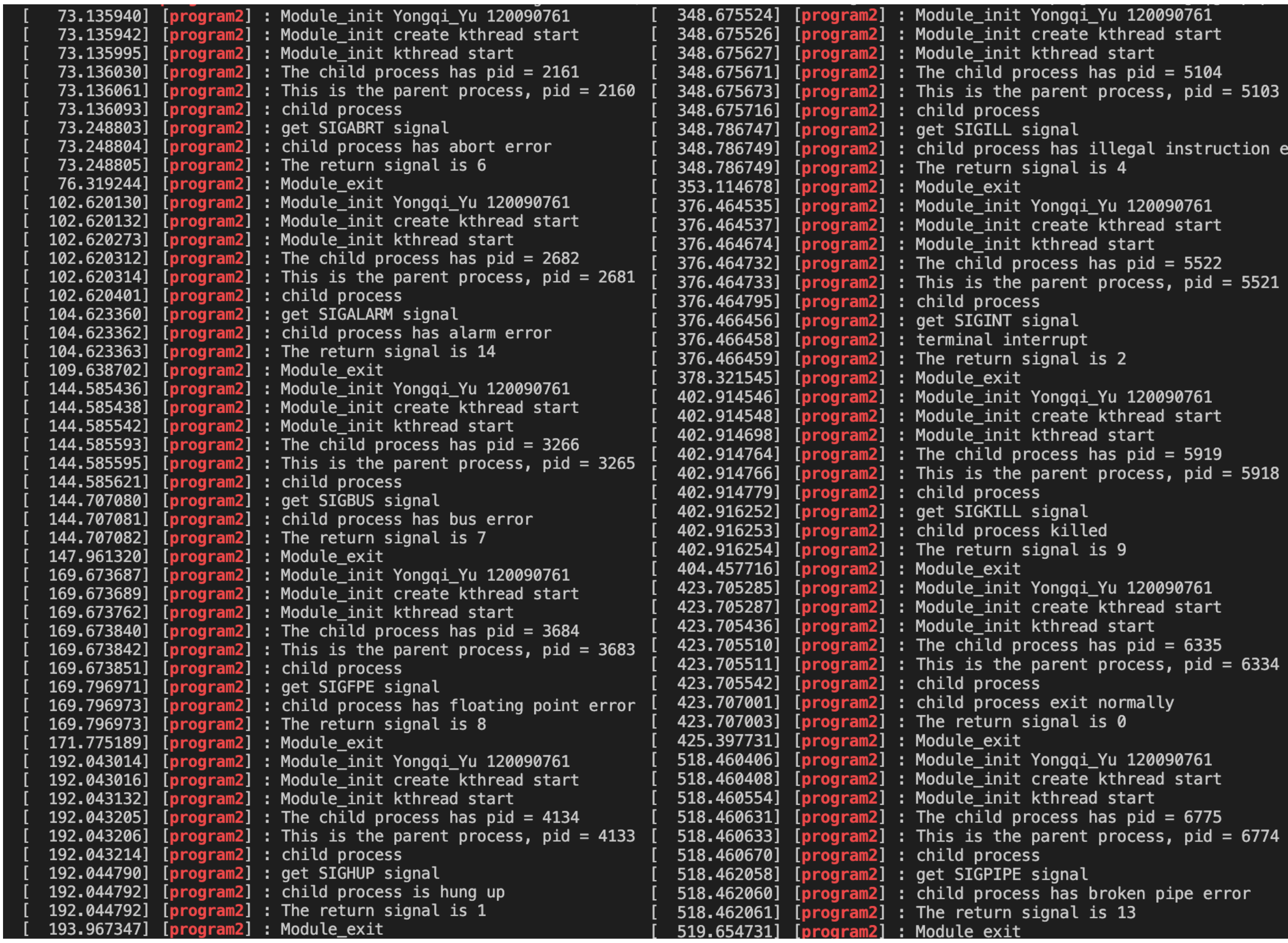
Task1



```
[  569.373901] [program2] : Module_init Yongqi_Yu 120090761
[  569.373903] [program2] : Module_init create kthread start
[  569.374020] [program2] : Module_init kthread start
[  569.374102] [program2] : The child process has pid = 7273
[  569.374103] [program2] : This is the parent process, pid = 7272
[  569.374122] [program2] : child process
[  569.478733] [program2] : get SIGQUIT signal
[  569.478735] [program2] : terminal quit
[  569.478735] [program2] : The return signal is 3
[  571.213035] [program2] : Module_exit
[  590.756760] [program2] : Module_init Yongqi_Yu 120090761
[  590.756762] [program2] : Module_init create kthread start
[  590.756892] [program2] : Module_init kthread start
[  590.756974] [program2] : The child process has pid = 7680
[  590.756976] [program2] : This is the parent process, pid = 7679
[  590.756985] [program2] : child process
[  590.864187] [program2] : get SIGSEGV signal
[  590.864188] [program2] : child process has segmentation fault err
[  590.864189] [program2] : The return signal is 11
[  592.563927] [program2] : Module_exit
[  611.835162] [program2] : Module_init Yongqi_Yu 120090761
[  611.835163] [program2] : Module_init create kthread start
[  611.835223] [program2] : Module_init kthread start
[  611.835292] [program2] : The child process has pid = 8088
[  611.835294] [program2] : This is the parent process, pid = 8087
[  611.835331] [program2] : child process
[  611.836680] [program2] : get SIGSTOP signal
[  611.836681] [program2] : child process stopped
[  611.836682] [program2] : The return signal is 19
[  613.482920] [program2] : Module_exit
[  629.195143] [program2] : Module_init Yongqi_Yu 120090761
[  629.195145] [program2] : Module_init create kthread start
[  629.195320] [program2] : Module_init kthread start
[  629.195719] [program2] : The child process has pid = 8514
[  629.195720] [program2] : This is the parent process, pid = 8512
[  629.195774] [program2] : child process
[  629.196830] [program2] : get SIGTERM signal
[  629.196831] [program2] : child process terminated
[  629.196832] [program2] : The return signal is 15
[  630.732766] [program2] : Module_exit
[  641.948284] [program2] : Module_init Yongqi_Yu 120090761
[  641.948286] [program2] : Module_init create kthread start
[  641.948398] [program2] : Module_init kthread start
[  641.948435] [program2] : The child process has pid = 8928
[  641.948436] [program2] : This is the parent process, pid = 8927
[  641.948463] [program2] : child process
[  642.057304] [program2] : get SIGTRAP signal
[  642.057305] [program2] : child process has trap error
[  642.057306] [program2] : The return signal is 5
[  643.499375] [program2] : Module_exit
```

Task2

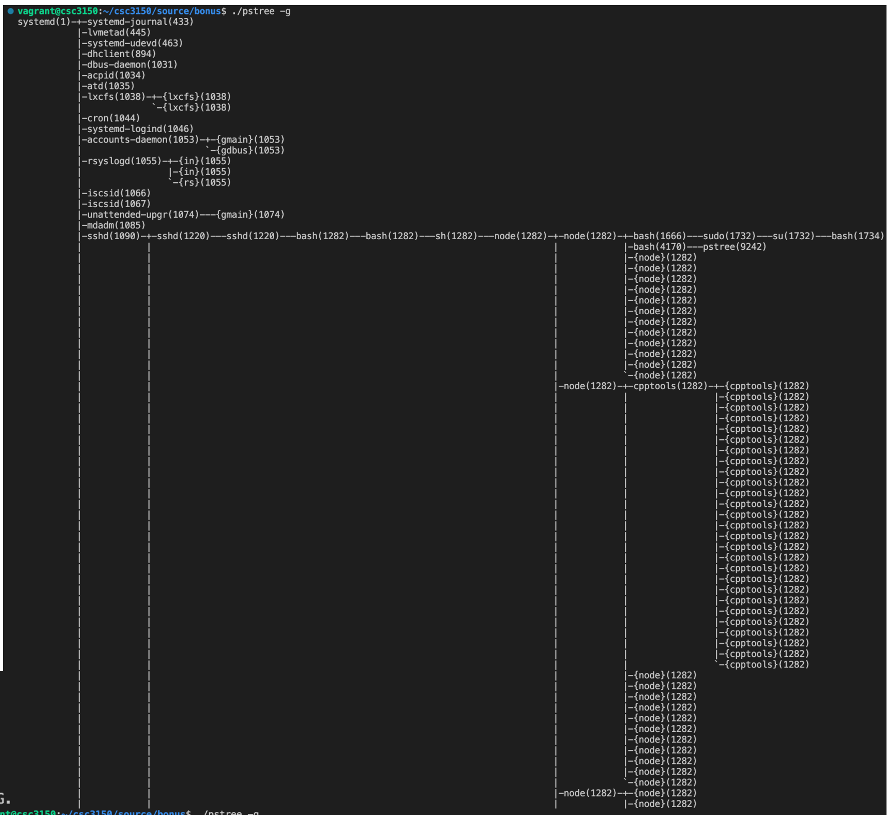
Task2



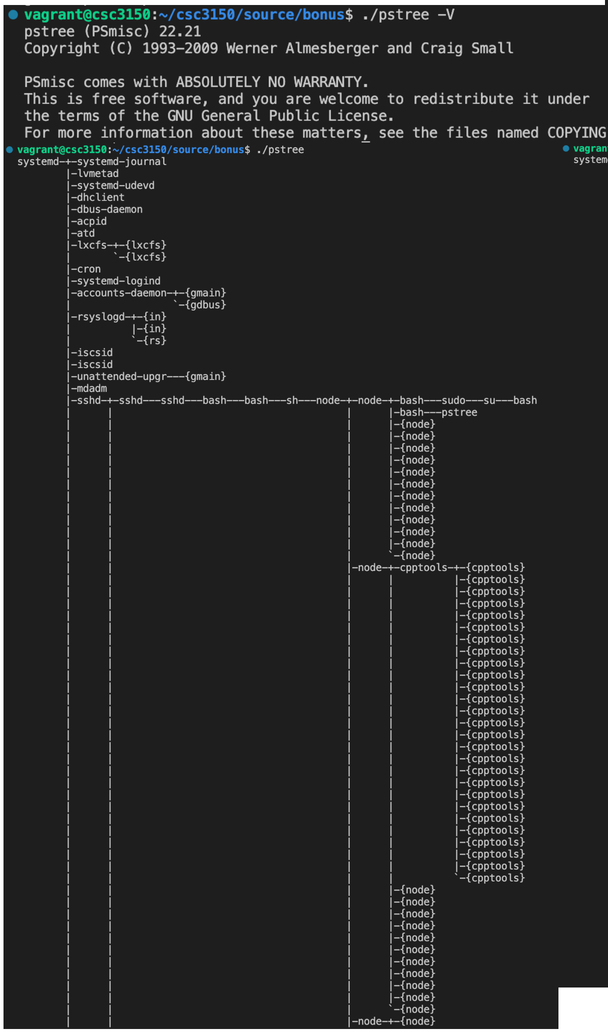
```
[   73.135940] [program2] : Module_init Yongqi_Yu 120090761
[   73.135942] [program2] : Module_init create kthread start
[   73.135995] [program2] : Module_init kthread start
[   73.136030] [program2] : The child process has pid = 2161
[   73.136061] [program2] : This is the parent process, pid = 2160
[   73.136093] [program2] : child process
[   73.248803] [program2] : get SIGABRT signal
[   73.248804] [program2] : child process has abort error
[   73.248805] [program2] : The return signal is 6
[   76.319244] [program2] : Module_exit
[  102.620130] [program2] : Module_init Yongqi_Yu 120090761
[  102.620132] [program2] : Module_init create kthread start
[  102.620273] [program2] : Module_init kthread start
[  102.620312] [program2] : The child process has pid = 2682
[  102.620314] [program2] : This is the parent process, pid = 2681
[  102.620401] [program2] : child process
[  104.623360] [program2] : get SIGALARM signal
[  104.623362] [program2] : child process has alarm error
[  104.623363] [program2] : The return signal is 14
[  109.638702] [program2] : Module_exit
[  144.585436] [program2] : Module_init Yongqi_Yu 120090761
[  144.585438] [program2] : Module_init create kthread start
[  144.585542] [program2] : Module_init kthread start
[  144.585593] [program2] : The child process has pid = 3266
[  144.585595] [program2] : This is the parent process, pid = 3265
[  144.585621] [program2] : child process
[  144.707080] [program2] : get SIGBUS signal
[  144.707081] [program2] : child process has bus error
[  144.707082] [program2] : The return signal is 7
[  147.961320] [program2] : Module_exit
[  169.673687] [program2] : Module_init Yongqi_Yu 120090761
[  169.673689] [program2] : Module_init create kthread start
[  169.673762] [program2] : Module_init kthread start
[  169.673840] [program2] : The child process has pid = 3684
[  169.673842] [program2] : This is the parent process, pid = 3683
[  169.673851] [program2] : child process
[  169.796971] [program2] : get SIGFPE signal
[  169.796973] [program2] : child process has floating point error
[  169.796973] [program2] : The return signal is 8
[  171.775189] [program2] : Module_exit
[  192.043014] [program2] : Module_init Yongqi_Yu 120090761
[  192.043016] [program2] : Module_init create kthread start
[  192.043132] [program2] : Module_init kthread start
[  192.043205] [program2] : The child process has pid = 4134
[  192.043206] [program2] : This is the parent process, pid = 4133
[  192.043214] [program2] : child process
[  192.044790] [program2] : get SIGHUP signal
[  192.044792] [program2] : child process is hung up
[  192.044792] [program2] : The return signal is 1
[  193.967347] [program2] : Module_exit
```

```
[  348.675524] [program2] : Module_init Yongqi_Yu 120090761
[  348.675526] [program2] : Module_init create kthread start
[  348.675627] [program2] : Module_init kthread start
[  348.675671] [program2] : The child process has pid = 5104
[  348.675673] [program2] : This is the parent process, pid = 5103
[  348.675716] [program2] : child process
[  348.786747] [program2] : get SIGILL signal
[  348.786749] [program2] : child process has illegal instruction e
[  348.786749] [program2] : The return signal is 4
[  353.114678] [program2] : Module_exit
[  376.464535] [program2] : Module_init Yongqi_Yu 120090761
[  376.464537] [program2] : Module_init create kthread start
[  376.464674] [program2] : Module_init kthread start
[  376.464732] [program2] : The child process has pid = 5522
[  376.464733] [program2] : This is the parent process, pid = 5521
[  376.464795] [program2] : child process
[  376.466456] [program2] : get SIGINT signal
[  376.466458] [program2] : terminal interrupt
[  376.466459] [program2] : The return signal is 2
[  378.321545] [program2] : Module_exit
[  402.914546] [program2] : Module_init Yongqi_Yu 120090761
[  402.914548] [program2] : Module_init create kthread start
[  402.914698] [program2] : Module_init kthread start
[  402.914764] [program2] : The child process has pid = 5919
[  402.914766] [program2] : This is the parent process, pid = 5918
[  402.914779] [program2] : child process
[  402.916252] [program2] : get SIGKILL signal
[  402.916253] [program2] : child process killed
[  402.916254] [program2] : The return signal is 9
[  404.457716] [program2] : Module_exit
[  423.705285] [program2] : Module_init Yongqi_Yu 120090761
[  423.705287] [program2] : Module_init create kthread start
[  423.705436] [program2] : Module_init kthread start
[  423.705510] [program2] : The child process has pid = 6335
[  423.705511] [program2] : This is the parent process, pid = 6334
[  423.705542] [program2] : child process
[  423.707001] [program2] : child process exit normally
[  423.707003] [program2] : The return signal is 0
[  425.397731] [program2] : Module_exit
[  518.460406] [program2] : Module_init Yongqi_Yu 120090761
[  518.460408] [program2] : Module_init create kthread start
[  518.460554] [program2] : Module_init kthread start
[  518.460631] [program2] : The child process has pid = 6775
[  518.460633] [program2] : This is the parent process, pid = 6774
[  518.460670] [program2] : child process
[  518.462058] [program2] : get SIGPIPE signal
[  518.462060] [program2] : child process has broken pipe error
[  518.462061] [program2] : The return signal is 13
[  519.654731] [program2] : Module_exit
```

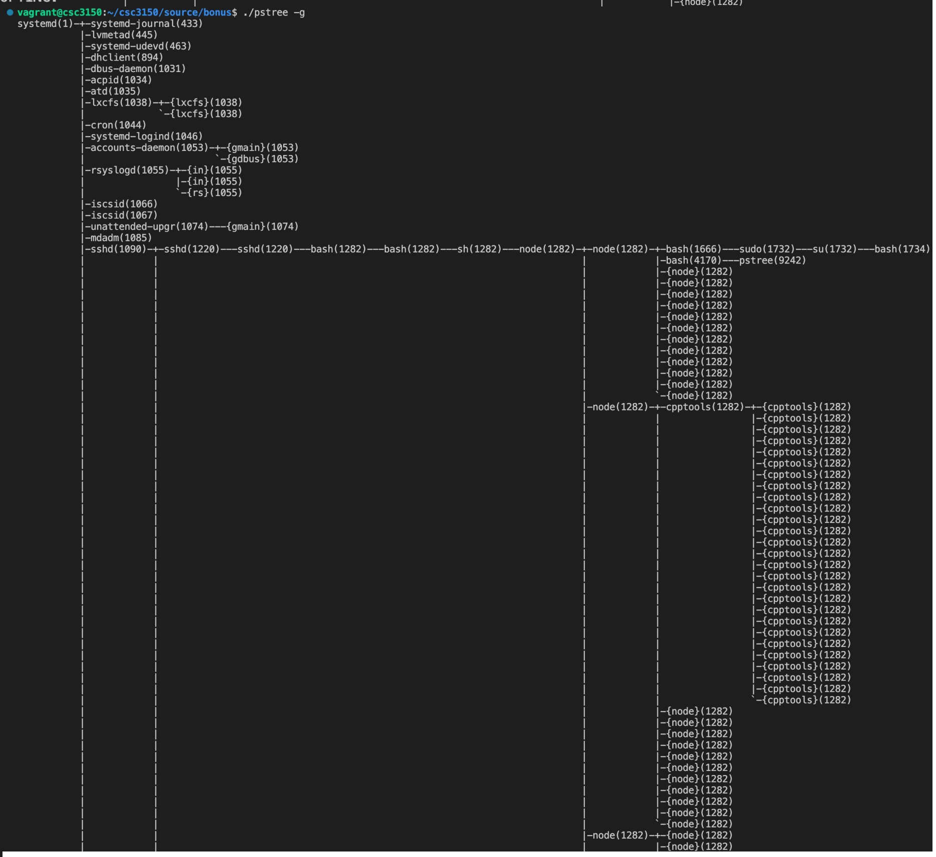
Bonus



```
● vagrant@csc3150:~/csc3150/source/bonus$ ./pstree -g
  systemd(1)-+-systemd-journal(433)
             |-lvmetad(445)
             |-systemd-udevd(463)
             |-dhclient(894)
             |-dbus-daemon(1031)
             |-acpid(1034)
             |-atd(1035)
             |-lxcfs(1038)-+-{lxcfs}(1038)
             |             `-{lxcfs}(1038)
             |-cron(1044)
             |-systemd-logind(1046)
             |-accounts-daemon(1053)-+-{gmain}(1053)
             |                       `-{gdbus}(1053)
             |-rsyslogd(1055)-+-{in}(1055)
             |                |-{in}(1055)
             |                `-{rs}(1055)
             |-iscsid(1066)
             |-iscsid(1067)
             |-unattended-upgr(1074)---{gmain}(1074)
             |-mdadm(1085)
             |-sshd(1090)-+-sshd(1220)---sshd(1220)---bash(1282)---bash(1282)---sh(1282)---node(1282)-+-node(1282)-+-bash(1666)---sudo(1732)---su(1732)---bash(1734)
             |            |                                                                           |            |-bash(4170)---pstree(9242)
             |            |                                                                           |            |-{node}(1282)
             |            |                                                                           |            |-{node}(1282)
             |            |                                                                           |            |-{node}(1282)
             |            |                                                                           |            |-{node}(1282)
             |            |                                                                           |            |-{node}(1282)
             |            |                                                                           |            |-{node}(1282)
             |            |                                                                           |            |-{node}(1282)
             |            |                                                                           |            |-{node}(1282)
             |            |                                                                           |            |-{node}(1282)
             |            |                                                                           |            |-{node}(1282)
             |            |                                                                           |            `-{node}(1282)
             |            |                                                                           |-node(1282)-+-cpptools(1282)-+-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                `-{cpptools}(1282)
             |            |                                                                           |            |-{node}(1282)
             |            |                                                                           |            |-{node}(1282)
             |            |                                                                           |            |-{node}(1282)
             |            |                                                                           |            |-{node}(1282)
             |            |                                                                           |            |-{node}(1282)
             |            |                                                                           |            |-{node}(1282)
             |            |                                                                           |            |-{node}(1282)
             |            |                                                                           |            |-{node}(1282)
             |            |                                                                           |            |-{node}(1282)
             |            |                                                                           |            |-{node}(1282)
             |            |                                                                           |            `-{node}(1282)
             |            |                                                                           |-node(1282)-+-{node}(1282)
             |            |                                                                           |            |-{node}(1282)
```



```
● vagrant@csc3150:~/csc3150/source/bonus$ ./pstree -V
  pstree (PSmisc) 22.21
  Copyright (C) 1993-2009 Werner Almesberger and Craig Small

  PSmisc comes with ABSOLUTELY NO WARRANTY.
  This is free software, and you are welcome to redistribute it under
  the terms of the GNU General Public License.
  For more information about these matters, see the files named COPYING.
● vagrant@csc3150:~/csc3150/source/bonus$ ./pstree
  systemd-+-systemd-journal
          |-lvmetad
          |-systemd-udevd
          |-dhclient
          |-dbus-daemon
          |-acpid
          |-atd
          |-lxcfs-+-{lxcfs}
          |       `-{lxcfs}
          |-cron
          |-systemd-logind
          |-accounts-daemon-+-{gmain}
          |                 `-{gdbus}
          |-rsyslogd-+-{in}
          |          |-{in}
          |          `-{rs}
          |-iscsid
          |-iscsid
          |-unattended-upgr---{gmain}
          |-mdadm
          |-sshd-+-sshd---sshd---bash---bash---sh---node-+-node-+-bash---sudo---su---bash
          |      |                                       |      |-bash---pstree
          |      |                                       |      |-{node}
          |      |                                       |      |-{node}
          |      |                                       |      |-{node}
          |      |                                       |      |-{node}
          |      |                                       |      |-{node}
          |      |                                       |      |-{node}
          |      |                                       |      |-{node}
          |      |                                       |      |-{node}
          |      |                                       |      |-{node}
          |      |                                       |      |-{node}
          |      |                                       |      |-{node}
          |      |                                       |      `-{node}
          |      |                                       |-node-+-cpptools-+-{cpptools}
          |      |                                       |      |          |-{cpptools}
          |      |                                       |      |          |-{cpptools}
          |      |                                       |      |          |-{cpptools}
          |      |                                       |      |          |-{cpptools}
          |      |                                       |      |          |-{cpptools}
          |      |                                       |      |          |-{cpptools}
          |      |                                       |      |          |-{cpptools}
          |      |                                       |      |          |-{cpptools}
          |      |                                       |      |          |-{cpptools}
          |      |                                       |      |          |-{cpptools}
          |      |                                       |      |          |-{cpptools}
          |      |                                       |      |          |-{cpptools}
          |      |                                       |      |          |-{cpptools}
          |      |                                       |      |          |-{cpptools}
          |      |                                       |      |          |-{cpptools}
          |      |                                       |      |          |-{cpptools}
          |      |                                       |      |          |-{cpptools}
          |      |                                       |      |          |-{cpptools}
          |      |                                       |      |          |-{cpptools}
          |      |                                       |      |          |-{cpptools}
          |      |                                       |      |          |-{cpptools}
          |      |                                       |      |          |-{cpptools}
          |      |                                       |      |          |-{cpptools}
          |      |                                       |      |          |-{cpptools}
          |      |                                       |      |          |-{cpptools}
          |      |                                       |      |          `-{cpptools}
          |      |                                       |      |-{node}
          |      |                                       |      |-{node}
          |      |                                       |      |-{node}
          |      |                                       |      |-{node}
          |      |                                       |      |-{node}
          |      |                                       |      |-{node}
          |      |                                       |      |-{node}
          |      |                                       |      |-{node}
          |      |                                       |      |-{node}
          |      |                                       |      |-{node}
          |      |                                       |      `-{node}
          |      |                                       |-node-+-{node}
```



```
● vagrant@csc3150:~/csc3150/source/bonus$ ./pstree -g
  systemd(1)-+-systemd-journal(433)
             |-lvmetad(445)
             |-systemd-udevd(463)
             |-dhclient(894)
             |-dbus-daemon(1031)
             |-acpid(1034)
             |-atd(1035)
             |-lxcfs(1038)-+-{lxcfs}(1038)
             |             `-{lxcfs}(1038)
             |-cron(1044)
             |-systemd-logind(1046)
             |-accounts-daemon(1053)-+-{gmain}(1053)
             |                       `-{gdbus}(1053)
             |-rsyslogd(1055)-+-{in}(1055)
             |                |-{in}(1055)
             |                `-{rs}(1055)
             |-iscsid(1066)
             |-iscsid(1067)
             |-unattended-upgr(1074)---{gmain}(1074)
             |-mdadm(1085)
             |-sshd(1090)-+-sshd(1220)---sshd(1220)---bash(1282)---bash(1282)---sh(1282)---node(1282)-+-node(1282)-+-bash(1666)---sudo(1732)---su(1732)---bash(1734)
             |            |                                                                           |            |-bash(4170)---pstree(9242)
             |            |                                                                           |            |-{node}(1282)
             |            |                                                                           |            |-{node}(1282)
             |            |                                                                           |            |-{node}(1282)
             |            |                                                                           |            |-{node}(1282)
             |            |                                                                           |            |-{node}(1282)
             |            |                                                                           |            |-{node}(1282)
             |            |                                                                           |            |-{node}(1282)
             |            |                                                                           |            |-{node}(1282)
             |            |                                                                           |            |-{node}(1282)
             |            |                                                                           |            |-{node}(1282)
             |            |                                                                           |            `-{node}(1282)
             |            |                                                                           |-node(1282)-+-cpptools(1282)-+-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                |-{cpptools}(1282)
             |            |                                                                           |            |                `-{cpptools}(1282)
             |            |                                                                           |            |-{node}(1282)
             |            |                                                                           |            |-{node}(1282)
             |            |                                                                           |            |-{node}(1282)
             |            |                                                                           |            |-{node}(1282)
             |            |                                                                           |            |-{node}(1282)
             |            |                                                                           |            |-{node}(1282)
             |            |                                                                           |            |-{node}(1282)
             |            |                                                                           |            |-{node}(1282)
             |            |                                                                           |            |-{node}(1282)
             |            |                                                                           |            |-{node}(1282)
             |            |                                                                           |            `-{node}(1282)
             |            |                                                                           |-node(1282)-+-{node}(1282)
             |            |                                                                           |            |-{node}(1282)
```

“-A”, “-c”, “-n”, “-l”will output the same result

What did I learn from the tasks?

In Task1, I learned how to fork a child process in user mode, how the child process executes files and return signals and how parent process wait the child process ends and receives its returned signal. Also, I learned how to handle the returned status, recognize its type and make further operations according to its type.

In Task2, I learned a lot of things. First, I got familiar with the process to install a kernel and compile it. Then, I knew how to change the kernel to open the function that can only be called in kernel mode. Also, I learned some basic operation of the kernel object(compiling, executing and so on). Then I learned how to fork a child process in kernel mode as well as the usage of some functions in the kernel, such as do_wait, do_execute, kernel_clone, etc..

In bonus, I learned about the /proc file system and got some knowledge about how to control the process of the computer. Also I tried to implement some complex data structure and get better know of the pointer in C language.

How to execute my programs?

Task1

1. make
2. ./program1 ./filename

Task2

1. sudo su
2. make
3. gcc -o test test.c
4. insmod program2.ko
5. rmmod program2.ko
6. dmesg |grep program2

Bonus

1. make
2. ./pstree [-argument]

Also in each folder you can type “make clean” to clean the binary file.