

## Program 1

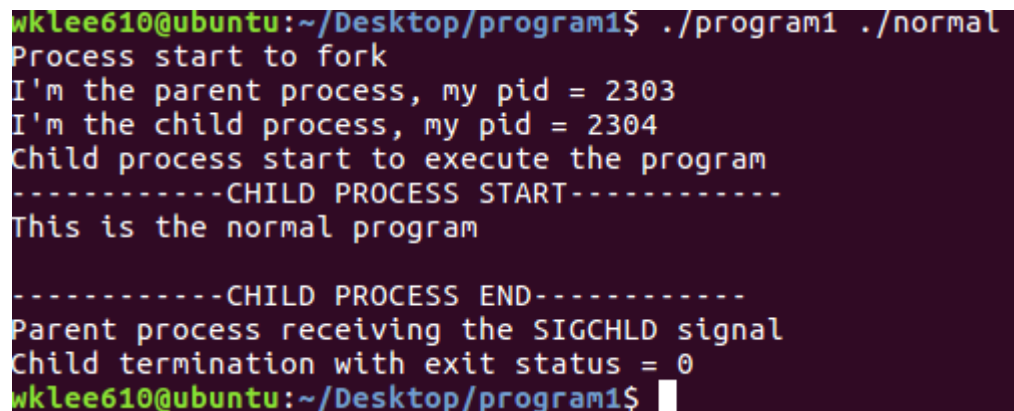
- How did you design your program?

Used fork() function to create child process in parent process. Used execv() function to execute the execution files from the child process. If the program can't find the file entered by the user, it will print error and directly break the child process. When file executed, parent process goes into 'wait' state and wait until child process ends. That's why I used waitpid() function in the code inside. Parent process has responsible for signals from the child process and whether to check child process is okay with the execution status. For wifexited(), wifstopped() functions, I used it for identify the status of child process and output different results.

- The steps to execute your program.

Used VM (Ubuntu16.04) and followed examples in the hw1 and tutorials.

- Screenshot of your program output.



```
wklee610@ubuntu:~/Desktop/program1$ ./program1 ./normal
Process start to fork
I'm the parent process, my pid = 2303
I'm the child process, my pid = 2304
Child process start to execute the program
-----CHILD PROCESS START-----
This is the normal program

-----CHILD PROCESS END-----
Parent process receiving the SIGCHLD signal
Child termination with exit status = 0
wklee610@ubuntu:~/Desktop/program1$
```

```
wklee610@ubuntu:~/Desktop/program1$ ./program1 ./terminate
Process start to fork
I'm the parent process, my pid = 2307
I'm the child process, my pid = 2308
Child process start to execute the program
-----CHILD PROCESS START-----
This is the SIGTERM program

Parent process receiving the SIGCHLD signal
child process get SIGTERM signal
child process is terminated by SIGTERM signal
CHILD EXECUTION FAILED!!
wklee610@ubuntu:~/Desktop/program1$
```

```
wklee610@ubuntu:~/Desktop/program1$ ./program1 ./stop
Process start to fork
I'm the parent process, my pid = 2309
I'm the child process, my pid = 2310
Child process start to execute the program
-----CHILD PROCESS START-----
This is the SIGSTOP program

Parent process receiving the SIGCHLD signal
child process stopped
CHILD PROXESS STOPPED
wklee610@ubuntu:~/Desktop/program1$
```

```
wklee610@ubuntu:~/Desktop/program1$ ./program1 ./alarm
Process start to fork
I'm the parent process, my pid = 2311
I'm the child process, my pid = 2312
Child process start to execute the program
-----CHILD PROCESS START-----
This is the SIGALRM program

Parent process receiving the SIGCHLD signal
child process get SIGALRM signal
child process is terminated by SIGALRM signal
CHILD EXECUTION FAILED!!
wklee610@ubuntu:~/Desktop/program1$
```

```
wklee610@ubuntu:~/Desktop/program1$ ./program1 ./abort
Process start to fork
I'm the parent process, my pid = 2315
I'm the child process, my pid = 2316
Child process start to execute the program
-----CHILD PROCESS START-----
This is the SIGABRT program

Parent process receiving the SIGCHLD signal
child process get SIGABRT signal
child process is terminated by SIGABRT signal
CHILD EXECUTION FAILED!!
wklee610@ubuntu:~/Desktop/program1$
```

```
wklee610@ubuntu:~/Desktop/program1$ ./program1 ./bus
Process start to fork
I'm the parent process, my pid = 2321
I'm the child process, my pid = 2322
Child process start to execute the program
-----CHILD PROCESS START-----
This is the SIGBUS program

Parent process receiving the SIGCHLD signal
child process get SIGBUS signal
child process is terminated by SIGBUS signal
CHILD EXECUTION FAILED!!
wklee610@ubuntu:~/Desktop/program1$
```

```
wklee610@ubuntu:~/Desktop/program1$ ./program1 ./floating
Process start to fork
I'm the parent process, my pid = 2326
I'm the child process, my pid = 2327
Child process start to execute the program
-----CHILD PROCESS START-----
This is the SIGFPE program

Parent process receiving the SIGCHLD signal
child process get SIGFPE signal
child process is terminated by SIGFPE signal
CHILD EXECUTION FAILED!!
wklee610@ubuntu:~/Desktop/program1$
```

```
wklee610@ubuntu:~/Desktop/program1$ ./program1 ./hangup
Process start to fork
I'm the parent process, my pid = 2329
I'm the child process, my pid = 2330
Child process start to execute the program
-----CHILD PROCESS START-----
This is the SIGHUP program

Parent process receiving the SIGCHLD signal
child process get SIGHUP signal
child process is terminated by SIGHUP signal
CHILD EXECUTION FAILED!!
wklee610@ubuntu:~/Desktop/program1$
```

```
wklee610@ubuntu:~/Desktop/program1$ ./program1 ./illegal_instr
Process start to fork
I'm the parent process, my pid = 2333
I'm the child process, my pid = 2334
Child process start to execute the program
-----CHILD PROCESS START-----
This is the SIGILL program

Parent process receiving the SIGCHLD signal
child process get SIGILL signal
child process is terminated by SIGILL signal
CHILD EXECUTION FAILED!!
wklee610@ubuntu:~/Desktop/program1$
```

```
wklee610@ubuntu:~/Desktop/program1$ ./program1 ./interrupt
Process start to fork
I'm the parent process, my pid = 2342
I'm the child process, my pid = 2343
Child process start to execute the program
-----CHILD PROCESS START-----
This is the SIGINT program

Parent process receiving the SIGCHLD signal
child process get SIGINT signal
child process is terminated by SIGINT signal
CHILD EXECUTION FAILED!!
wklee610@ubuntu:~/Desktop/program1$
```

```
wklee610@ubuntu:~/Desktop/program1$ ./program1 ./kill
Process start to fork
I'm the parent process, my pid = 2346
I'm the child process, my pid = 2347
Child process start to execute the program
-----CHILD PROCESS START-----
This is the SIGKILL program

Parent process receiving the SIGCHLD signal
child process get SIGKILL signal
child process is terminated by SIGKILL signal
CHILD EXECUTION FAILED!!
wklee610@ubuntu:~/Desktop/program1$
```

```
wklee610@ubuntu:~/Desktop/program1$ ./program1 ./pipe
Process start to fork
I'm the parent process, my pid = 2352
I'm the child process, my pid = 2353
Child process start to execute the program
-----CHILD PROCESS START-----
This is the SIGPIPE program

Parent process receiving the SIGCHLD signal
child process get SIGPIPE signal
child process is terminated by SIGPIPE signal
CHILD EXECUTION FAILED!!
wklee610@ubuntu:~/Desktop/program1$
```

```
wklee610@ubuntu:~/Desktop/program1$ ./program1 ./quit
Process start to fork
I'm the parent process, my pid = 2354
I'm the child process, my pid = 2355
Child process start to execute the program
-----CHILD PROCESS START-----
This is the SIGQUIT program

Parent process receiving the SIGCHLD signal
child process get SIGQUIT signal
child process is terminated by SIGQUIT signal
CHILD EXECUTION FAILED!!
wklee610@ubuntu:~/Desktop/program1$
```

```
wklee610@ubuntu:~/Desktop/program1$ ./program1 ./segment_fault
Process start to fork
I'm the parent process, my pid = 2363
I'm the child process, my pid = 2364
Child process start to execute the program
-----CHILD PROCESS START-----
This is the SIGSEGV program

Parent process receiving the SIGCHLD signal
child process get SIGSEGV signal
child process is terminated by SIGSEGV signal
CHILD EXECUTION FAILED!!
wklee610@ubuntu:~/Desktop/program1$
```

```
CHILD EXECUTION FAILED!!
wklee610@ubuntu:~/Desktop/program1$ ./program1 ./stop
Process start to fork
I'm the parent process, my pid = 2368
I'm the child process, my pid = 2369
Child process start to execute the program
-----CHILD PROCESS START-----
This is the SIGSTOP program

Parent process receiving the SIGCHLD signal
child process stopped
CHILD PROXESS STOPPED
wklee610@ubuntu:~/Desktop/program1$
```

```
wklee610@ubuntu:~/Desktop/program1$ ./program1 ./trap
Process start to fork
I'm the parent process, my pid = 2372
I'm the child process, my pid = 2373
Child process start to execute the program
-----CHILD PROCESS START-----
This is the SIGTRAP program

Parent process receiving the SIGCHLD signal
child process get SIGTRAP signal
child process is terminated by SIGTRAP signal
CHILD EXECUTION FAILED!!
wklee610@ubuntu:~/Desktop/program1$
```



## Program 2

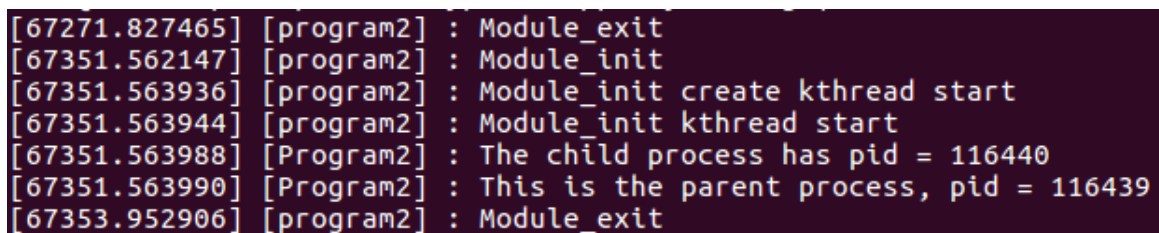
- How did you design your program?

Export `_do_fork()`, `do_wait()`, `do_execve()` and `getname()` functions and recompile the kernel and externed functions. I used `kthread_run()` to create new kernel thread. I wrote 'myfork' function directly execute in the thread and then use `_do_fork()` to fork a process. I made the new child process execute another function called `my_exec()`. I wrote a function called `my_wait()` to make the parent process wait for the end of the child process. I used `do_wait()` to determine the type of signal, check the variable called `wo.wo_stat` in the return result of `wo`. Last, use `printk()` for output.

- The steps to execute your program.

1. 'Sudo su' command for root
2. Go to directory and use 'make' command for compile.
3. Use 'insmod program2.ko' command to input kernel modules
4. use 'dmesg' command to see result.
5. Last, use 'rmmod program2' to delete module.

- Screenshot of your program output.



```
[67271.827465] [program2] : Module_exit
[67351.562147] [program2] : Module_init
[67351.563936] [program2] : Module_init create kthread start
[67351.563944] [program2] : Module_init kthread start
[67351.563988] [Program2] : The child process has pid = 116440
[67351.563990] [Program2] : This is the parent process, pid = 116439
[67353.952906] [program2] : Module_exit
```

After complete this project, I'm pretty familiar with processes and threads. I understood how to create VM and knew how to fork new processes. Before, I haven't know about Linux command, kernel space and processes clearly, but now after doing this project, much familiar with Linux command as well.

