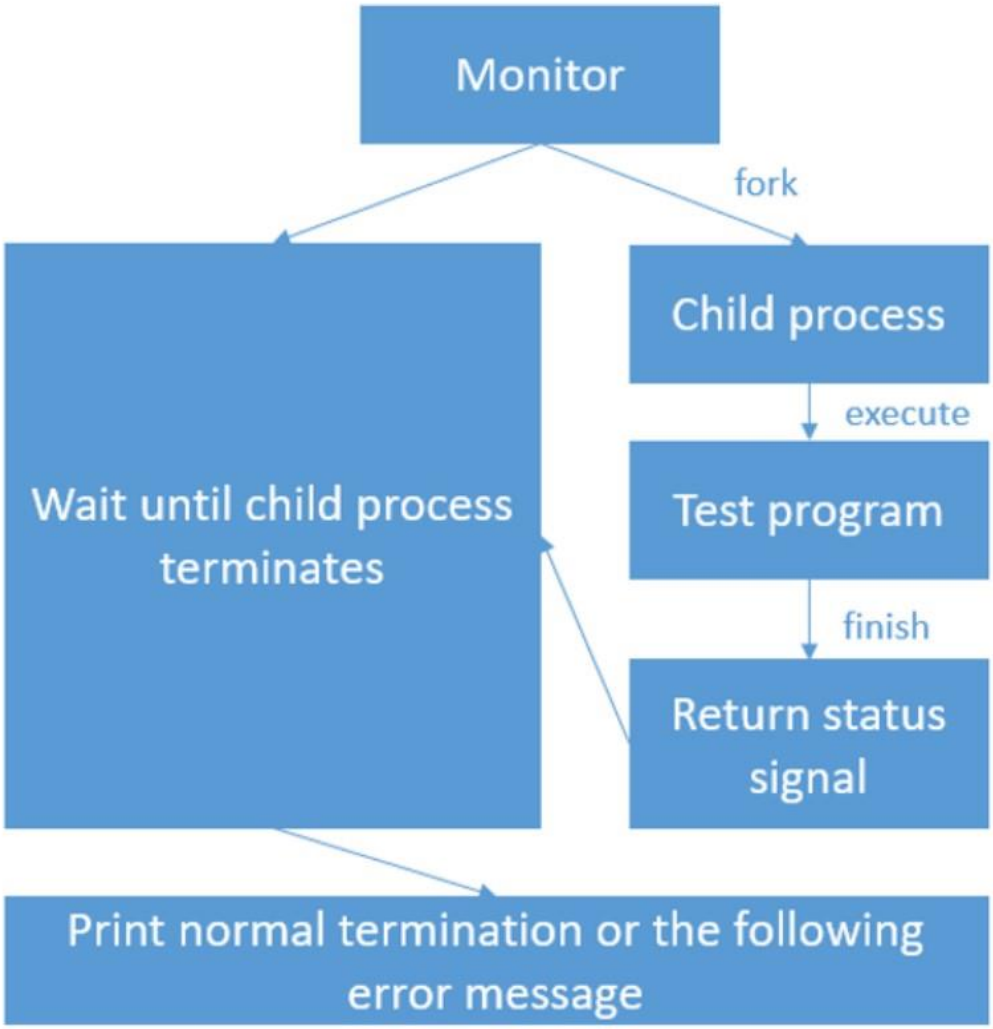
A1_Report

Zhu, Shenghao_120090548

Q: How did you design your program?

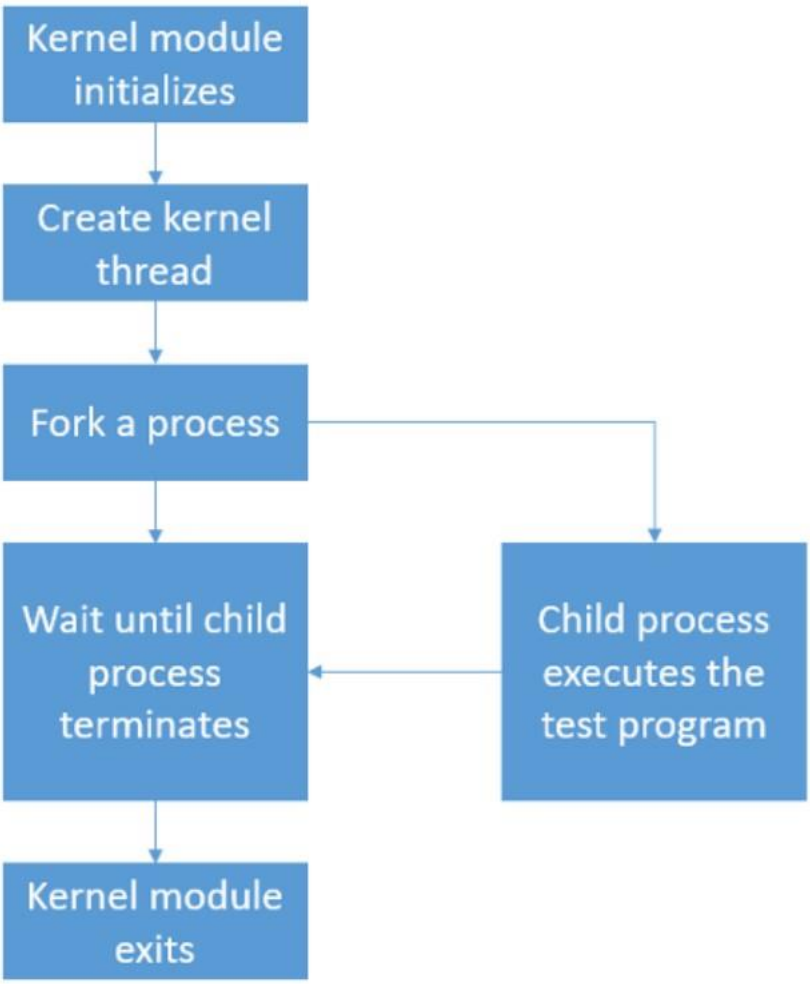
A: The program_1 chart flow is:



We first use fork() function to fork a child process, and then we use execve() function to execute the test program. We use waitpid() function to make parent process wait the child process until it terminate and send some siganls back. In summary, 1. Fork a child process to execute test program. 2. Use waitpid() to let the parent process receives the SIGCHLD signal. 3. Print out the termination information of child

process.

The program_2 char flow is:

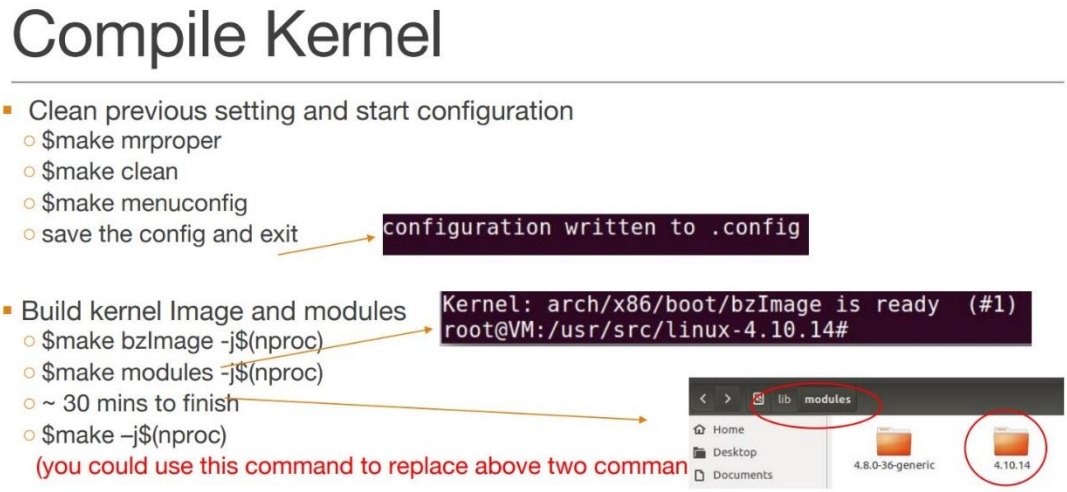


We need to fork the process in the kernel code. First we need to make the kernel and kernel modules in the computer before we type make command. program_init() function is to initialize the kernel and create the kernel thread. my_exec() function is used to test the kernel. my_fork() function is to fork the process and get the pid for parent and child process. my_wait function is used to wait for the child process terminate and send some signal to parent. kernel_clone() function is to allocate a new process resources area in my fork with given function names and other

parameters. my_exec() function, called by kernel_clone() function, will come in and start the execution of another test file. my_exec() function takes care of locating the test file, passing in the arguments and starts the execution. my_wait() function do its job in the parent process. Within the my_wait() function, struct wait_opts is constructed and passed to do_wait() function 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() function 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 related log. 6. Recompile the Linux kernel source code to use its function.

Q: How to set up your development environment, including how to compile kernel?

A: First update kernel version to 5.10.146.



Compile Kernel

- Install kernel modules
  - $make modules_install

```
  DEPMOD  4.10.14
root@VM:/home/seed/sdb4/linux-4.10.14#
```

- Install kernel
  - $make install

```
done
root@VM:/home/seed/sdb4/linux-4.10.14#
```

- Reboot to load new kernel
  - $reboot

  (When rebooting, you should select the updated kernel)

To finish program_2, we need to export do_execve(), do_wait(), kernel_clone() and getname_kernel() under the kernel source code.

```
static int do_execve(struct filename *filename,
    const char __user *const __user *__argv,
    const char __user *const __user *__envp)
{
    struct user_arg_ptr argv = { .ptr.native = __argv };
    struct user_arg_ptr envp = { .ptr.native = __envp };
    return do_execveat_common(AT_FDCWD, filename, argv, envp, 0);
}
EXPORT_SYMBOL(do_execve);
```

```
    wake_up_new_task(p);

    /* forking complete and child started to run, tell ptracer */
    if (unlikely(trace))
        ptrace_event_pid(trace, pid);

    if (clone_flags & CLONE_VFORK) {
        if (!wait_for_vfork_done(p, &vfork))
            ptrace_event_pid(PTRACE_EVENT_VFORK_DONE, pid);
    }

    put_pid(pid);
    return nr;
}
EXPORT_SYMBOL(kernel_clone);
```

```
        retval = -ERESTARTSYS;
        if (!signal_pending(current)) {
            schedule();
            goto repeat;
        }
    }
end:
    __set_current_state(TASK_RUNNING);
    remove_wait_queue(&current->signal->wait_chldexit, &wo->child_wait);
    return retval;
}
EXPORT_SYMBOL(do_wait);
```

```
        result = tmp;
    } else {
        __putname(result);
        return ERR_PTR(-ENAMETOOLONG);
    }
    memcpy((char *)result->name, filename, len);
    result->uptr = NULL;
    result->aname = NULL;
    result->refcnt = 1;
    audit_getname(result);

    return result;
}
EXPORT_SYMBOL(getname_kernel);
```

Then we recompile the kernel, starting from $make bzImage -j$(nproc).

Compile Kernel

- Install kernel modules
  - $make modules_install

```
  DEPMOD  4.10.14
root@VM:/home/seed/sdb4/linux-4.10.14#
```

- Install kernel
  - $make install

```
done
root@VM:/home/seed/sdb4/linux-4.10.14#
```

- Reboot to load new kernel
  - $reboot

  (When rebooting, you should select the updated kernel)

Therefore, we could extern those functions above and use those functions in our program.

Q: Screenshot of your program output.

A:Program1:

```
vagrant@csc3150:~/Assignment1/program1$ ./program1 ./abort
Process statrt to fork
I'm the Parent Process, my pid = 3611
I'm the Child Process, my pid = 3612
Child process start to execute the program:
------------CHILD PROCESS START------------
This is the SIGABRT program

Parent Process receives the SIGCHLD signal
Child process is aborted by abort signal
SIGABRT signal was raised in child process
```

```
vagrant@csc3150:~/Assignment1/program1$ ./program1 ./alarm
Process statrt to fork
I'm the Parent Process, my pid = 3666
I'm the Child Process, my pid = 3667
Child process start to execute the program:
------------CHILD PROCESS START------------
This is the SIGALRM program

Parent Process receives the SIGCHLD signal
Child process is aborted by alarm signal
SIGALRM signal was raised in child process
```

```
vagrant@csc3150:~/Assignment1/program1$ ./program1 ./bus
Process statrt to fork
I'm the Parent Process, my pid = 3736
I'm the Child Process, my pid = 3737
Child process start to execute the program:
------------CHILD PROCESS START------------
This is the SIGBUS program

Parent Process receives the SIGCHLD signal
Child process is aborted by bus signal
SIGBUS signal was raised in child process
```

```
vagrant@csc3150:~/Assignment1/program1$ ./program1 ./floating
Process statrt to fork
I'm the Parent Process, my pid = 3856
I'm the Child Process, my pid = 3857
Child process start to execute the program:
------------CHILD PROCESS START------------
This is the SIGFPE program

Parent Process receives the SIGCHLD signal
Child process is aborted by SIGFPE signal
SIGFPE signal was raised in child process
```

```
vagrant@csc3150:~/Assignment1/program1$ ./program1 ./hangup
Process statrt to fork
I'm the Parent Process, my pid = 3895
I'm the Child Process, my pid = 3896
Child process start to execute the program:
------------CHILD PROCESS START------------
This is the SIGHUP program

Parent Process receives the SIGCHLD signal
Child process is hung up
SIGHUP signal was raised in child process
```

```
vagrant@csc3150:~/Assignment1/program1$ ./program1 ./illegal_instr
Process statrt to fork
I'm the Parent Process, my pid = 3958
I'm the Child Process, my pid = 3959
Child process start to execute the program:
------------CHILD PROCESS START------------
This is the SIGILL program

Parent Process receives the SIGCHLD signal
Child process is aborted by SIGILL signal
SIGILL signal was raised in child process
```

```
vagrant@csc3150:~/Assignment1/program1$ ./program1 ./interrupt
Process statrt to fork
I'm the Parent Process, my pid = 3997
I'm the Child Process, my pid = 3998
Child process start to execute the program:
------------CHILD PROCESS START------------
This is the SIGINT program

Parent Process receives the SIGCHLD signal
Child process is aborted by SIGINT signal
SIGINT signal was raised in child process
```

```
vagrant@csc3150:~/Assignment1/program1$ ./program1 ./kill
Process statrt to fork
I'm the Parent Process, my pid = 4011
I'm the Child Process, my pid = 4012
Child process start to execute the program:
------------CHILD PROCESS START------------
This is the SIGKILL program

Parent Process receives the SIGCHLD signal
Child process is aborted by SIGKILL signal
SIGKILL signal was raised in child process
```

```
vagrant@csc3150:~/Assignment1/program1$ ./program1 ./normal
Process statrt to fork
I'm the Parent Process, my pid = 4065
I'm the Child Process, my pid = 4066
Child process start to execute the program:
------------CHILD PROCESS START------------
This is the normal program

------------CHILD PROCESS END------------
Parent Process receives the SIGCHLD signal
Normal termination with EXIT STATUS = 0
```

```
vagrant@csc3150:~/Assignment1/program1$ ./program1 ./pipe
Process statrt to fork
I'm the Parent Process, my pid = 4115
I'm the Child Process, my pid = 4116
Child process start to execute the program:
------------CHILD PROCESS START------------
This is the SIGPIPE program

Parent Process receives the SIGCHLD signal
Child process is aborted by SIGPIPE signal
SIGPIPE signal was raised in child process
```

```
vagrant@csc3150:~/Assignment1/program1$ ./program1 ./quit
Process statrt to fork
I'm the Parent Process, my pid = 4169
I'm the Child Process, my pid = 4170
Child process start to execute the program:
------------CHILD PROCESS START------------
This is the SIGQUIT program

Parent Process receives the SIGCHLD signal
Child process is aborted by SIGQUIT signal
SIGQUIT signal was raised in child process
```

```
vagrant@csc3150:~/Assignment1/program1$ ./program1 ./segment_fault
Process statrt to fork
I'm the Parent Process, my pid = 4231
I'm the Child Process, my pid = 4232
Child process start to execute the program:
------------CHILD PROCESS START------------
This is the SIGSEGV program

Parent Process receives the SIGCHLD signal
Child process is aborted by SIGSEGV signal
SIGSEGV signal was raised in child process
```

```
vagrant@csc3150:~/Assignment1/program1$ ./program1 ./stop
Process statrt to fork
I'm the Parent Process, my pid = 4261
I'm the Child Process, my pid = 4262
Child process start to execute the program:
------------CHILD PROCESS START------------
This is the SIGSTOP program

Parent Process receives the SIGCHLD signal
CHILD PROCESS STOPPED
```

```
vagrant@csc3150:~/Assignment1/program1$ ./program1 ./terminate
Process statrt to fork
I'm the Parent Process, my pid = 4290
I'm the Child Process, my pid = 4291
Child process start to execute the program:
------------CHILD PROCESS START------------
This is the SIGTERM program

Parent Process receives the SIGCHLD signal
Child process is aborted by SIGTERM signal
SIGTERM signal was raised in child process
```

```
vagrant@csc3150:~/Assignment1/program1$ ./program1 ./trap
Process statrt to fork
I'm the Parent Process, my pid = 4327
I'm the Child Process, my pid = 4328
Child process start to execute the program:
------------CHILD PROCESS START------------
This is the SIGTRAP program

Parent Process receives the SIGCHLD signal
Child process is aborted by trap signal
SIGTRAP signal was raised in child process
```

Program2:

```
root@csc3150:/home/vagrant/Assignment1/program2# dmesg | tail
[45561.353550] [program2] : Module_init {Zhu, Shenghao} {120090548}
[45561.593075] [program2] : Module_init create kthread start
[45561.631505] [program2] : Module_init kthread start
[45561.660114] [program2] : The child process has pid 19861
[45561.675739] [program2] : The parent process has pid 19860
[45561.700182] [program2] : child process
[45561.948174] [program2] : get SIGABRT signal
[45562.007889] [program2] : child process terminated
[45562.049017] [program2] : The return signal is 6
[45565.737034] [program2] : Module_exit
```

```
root@csc3150:/home/vagrant/Assignment1/program2# dmesg | tail
[45721.114319] [program2] : Module_init {Zhu, Shenghao} {120090548}
[45721.179405] [program2] : Module_init create kthread start
[45721.255602] [program2] : Module_init kthread start
[45721.304792] [program2] : The child process has pid 20724
[45721.322449] [program2] : The parent process has pid 20723
[45721.339232] [program2] : child process
[45723.308379] [program2] : get SIGALARM signal
[45723.328605] [program2] : child process terminated
[45723.341917] [program2] : The return signal is 14
[45725.986034] [program2] : Module_exit
```

```
root@csc3150:/home/vagrant/Assignment1/program2# dmesg | tail
[45842.942532] [program2] : Module_init {Zhu, Shenghao} {120090548}
[45843.011438] [program2] : Module_init create kthread start
[45843.082205] [program2] : Module_init kthread start
[45843.128532] [program2] : The child process has pid 21414
[45843.166401] [program2] : The parent process has pid 21413
[45843.220855] [program2] : child process
[45843.462374] [program2] : get SIGBUS signal
[45843.520432] [program2] : child process terminated
[45843.560609] [program2] : The return signal is 7
[45847.756634] [program2] : Module_exit
```

```
root@csc3150:/home/vagrant/Assignment1/program2# dmesg | tail
[45959.331629] [program2] : Module_init {Zhu, Shenghao} {120090548}
[45959.363849] [program2] : Module_init create kthread start
[45959.437356] [program2] : Module_init kthread start
[45959.474188] [program2] : The child process has pid 22910
[45959.580546] [program2] : The parent process has pid 22909
[45959.728405] [program2] : child process
[45959.728613] [program2] : get SIGFPE signal
[45959.790096] [program2] : child process terminated
[45959.832280] [program2] : The return signal is 8
[45964.792677] [program2] : Module_exit
```

```
root@csc3150:/home/vagrant/Assignment1/program2# dmesg | tail
[46066.846958] [program2] : Module_init {Zhu, Shenghao} {120090548}
[46066.862155] [program2] : Module_init create kthread start
[46066.878927] [program2] : Module_init kthread start
[46066.907255] [program2] : The child process has pid 23784
[46066.918633] [program2] : The parent process has pid 23783
[46066.947965] [program2] : child process
[46066.947983] [program2] : get SIGHUP signal
[46067.028067] [program2] : child process terminated
[46067.055788] [program2] : The return signal is 1
[46070.935816] [program2] : Module_exit
```

```
root@csc3150:/home/vagrant/Assignment1/program2# dmesg | tail
[46137.668044] [program2] : Module_init {Zhu, Shenghao} {120090548}
[46137.755761] [program2] : Module_init create kthread start
[46137.850862] [program2] : Module_init kthread start
[46137.905464] [program2] : The child process has pid 24510
[46137.917429] [program2] : The parent process has pid 24509
[46137.940333] [program2] : child process
[46138.198111] [program2] : get SIGILL signal
[46138.253896] [program2] : child process terminated
[46138.300670] [program2] : The return signal is 4
[46152.799028] [program2] : Module_exit
```

```
root@csc3150:/home/vagrant/Assignment1/program2# dmesg | tail
[46249.941483] [program2] : Module_init {Zhu, Shenghao} {120090548}
[46249.989442] [program2] : Module_init create kthread start
[46250.036028] [program2] : Module_init kthread start
[46250.077241] [program2] : The child process has pid 25241
[46250.092027] [program2] : The parent process has pid 25240
[46250.107141] [program2] : child process
[46250.107152] [program2] : get SIGINT signal
[46250.127957] [program2] : child process terminated
[46250.141344] [program2] : The return signal is 2
[46254.860296] [program2] : Module_exit
```

```
root@csc3150:/home/vagrant/Assignment1/program2# dmesg | tail
[46343.668100] [program2] : Module_init {Zhu, Shenghao} {120090548}
[46343.714276] [program2] : Module_init create kthread start
[46343.754078] [program2] : Module_init kthread start
[46343.769432] [program2] : The child process has pid 26012
[46343.983219] [program2] : The parent process has pid 26011
[46344.013448] [program2] : child process
[46344.013468] [program2] : get SIGKILL signal
[46344.086567] [program2] : child process terminated
[46344.124064] [program2] : The return signal is 9
[46347.752017] [program2] : Module_exit
```

```
root@csc3150:/home/vagrant/Assignment1/program2# dmesg | tail
[46347.752017] [program2] : Module_exit
[46407.638258] [program2] : Module_init {Zhu, Shenghao} {120090548}
[46407.706825] [program2] : Module_init create kthread start
[46407.778533] [program2] : Module_init kthread start
[46407.832330] [program2] : The child process has pid 26722
[46407.864175] [program2] : The parent process has pid 26721
[46407.891547] [program2] : child process
[46407.891581] [program2] : child process exit normally
[46407.964985] [program2] : The return signal is 0
[46413.704133] [program2] : Module_exit
```

```
root@csc3150:/home/vagrant/Assignment1/program2# dmesg | tail
[46475.740372] [program2] : Module_init {Zhu, Shenghao} {120090548}
[46475.782116] [program2] : Module_init create kthread start
[46475.846312] [program2] : Module_init kthread start
[46475.874470] [program2] : The child process has pid 27492
[46475.899914] [program2] : The parent process has pid 27491
[46475.932093] [program2] : child process
[46475.932115] [program2] : get SIGPIPE signal
[46476.000202] [program2] : child process terminated
[46476.046989] [program2] : The return signal is 13
[46481.315771] [program2] : Module_exit
```

```
root@csc3150:/home/vagrant/Assignment1/program2# dmesg | tail
[46550.266110] [program2] : Module_init {Zhu, Shenghao} {120090548}
[46550.317422] [program2] : Module_init create kthread start
[46550.374916] [program2] : Module_init kthread start
[46550.422517] [program2] : The child process has pid 28182
[46550.437366] [program2] : The parent process has pid 28181
[46550.490682] [program2] : child process
[46550.694783] [program2] : get SIGQUIT signal
[46550.753451] [program2] : child process terminated
[46550.792292] [program2] : The return signal is 3
[46554.188337] [program2] : Module_exit
```

```
root@csc3150:/home/vagrant/Assignment1/program2# dmesg | tail
[46623.510685] [program2] : Module_init {Zhu, Shenghao} {120090548}
[46623.564467] [program2] : Module_init create kthread start
[46623.616653] [program2] : Module_init kthread start
[46623.655010] [program2] : The child process has pid 28904
[46623.672889] [program2] : The parent process has pid 28903
[46623.868854] [program2] : child process
[46623.914318] [program2] : get SIGSEGV signal
[46623.970326] [program2] : child process terminated
[46624.008396] [program2] : The return signal is 11
[46627.926928] [program2] : Module_exit
```

```
root@csc3150:/home/vagrant/Assignment1/program2# dmesg | tail
[48627.837221] [program2] : Module_init {Zhu, Shenghao} {120090548}
[48627.851808] [program2] : Module_init create kthread start
[48627.867367] [program2] : Module_init kthread start
[48627.879851] [program2] : The child process has pid 2075
[48628.221926] [program2] : The parent process has pid 2074
[48628.239366] [program2] : child process
[48628.239394] [program2] : get SIGTERM signal
[48628.303834] [program2] : child process terminated
[48628.342456] [program2] : The return signal is 15
[48632.623966] [program2] : Module_exit
```

```
root@csc3150:/home/vagrant/Assignment1/program2# dmesg | tail
[48709.616554] [program2] : Module_init {Zhu, Shenghao} {120090548}
[48709.681644] [program2] : Module_init create kthread start
[48709.738960] [program2] : Module_init kthread start
[48709.789631] [program2] : The child process has pid 2843
[48709.807935] [program2] : The parent process has pid 2841
[48709.855406] [program2] : child process
[48710.093335] [program2] : get SIGTRAP signal
[48710.176599] [program2] : child process terminated
[48710.233775] [program2] : The return signal is 5
[48713.712197] [program2] : Module_exit
```

Q: What did you learn from the tasks?

A: In this program, I learnt how to fork child processes and print their related information. Also, I learnt how to fork the child process in the kernel space.