Report (120090671 Yuzheng Cong)

1. Program1

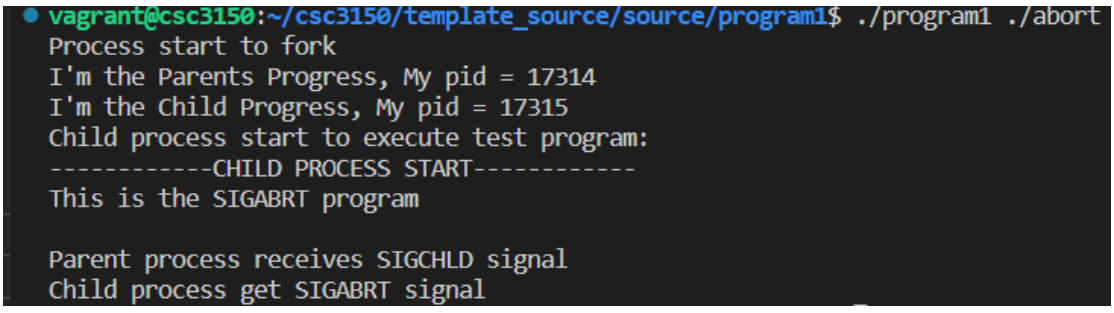
Program1 is a simulation of forking a process. When the monitor forks a child process to execute the test program, the process may encounter some other signals which stop the process from outputting the normal termination.

1.1 Implementation

Fork function is used to create the process. If fork returns 0, it means a new child process is created. If fork returns a positive value, the process ID of the child process is passed to parent. Getpid function is used to get the pid of the child process and the parent process. WIFEXIED, WIFSIGNALED, WIFSTOPPED can analyze the status referenced by the status argument.

1.2 Output

1. abort



2. alarm

```
vagrant@csc3150:~/csc3150/template_source/source/program1$ ./program1 ./alarm
Process start to fork
I'm the Parents Progress, My pid = 17330
I'm the Child Progress, My pid = 17331
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGALRM program

Parent process receives SIGCHLD signal
Child process get SIGALRM signal
```

3. bus

```
vagrant@csc3150:~/csc3150/template_source/source/program1$ ./program1 ./bus
Process start to fork
I'm the Parents Progress, My pid = 17372
I'm the Child Progress, My pid = 17373
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
```

4. floating

```
vagrant@csc3150:~/csc3150/template_source/source/program1$ ./program1 ./floating
Process start to fork
I'm the Parents Progress, My pid = 17408
I'm the Child Progress, My pid = 17409
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
```

5. hangup

```
vagrant@csc3150:~/csc3150/template_source/source/program1$ ./program1 ./hangup
Process start to fork
I'm the Parents Progress, My pid = 17450
I'm the Child Progress, My pid = 17451
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
```

6. illegal_instr

```
vagrant@csc3150:~/csc3150/template_source/source/program1$ ./program1 ./illegal_instr
Process start to fork
I'm the Parents Progress, My pid = 17498
I'm the Child Progress, My pid = 17499
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
```

7. interrupt

```
vagrant@csc3150:~/csc3150/template_source/source/program1$ ./program1 ./interrupt
Process start to fork
I'm the Parents Progress, My pid = 17549
I'm the Child Progress, My pid = 17550
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
```

8. kill

```
vagrant@csc3150:~/csc3150/template_source/source/program1$ ./program1 ./kill
Process start to fork
I'm the Parents Progress, My pid = 17578
I'm the Child Progress, My pid = 17579
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
```

9. normal

```
vagrant@csc3150:~/csc3150/template_source/source/program1$ ./program1 ./normal
Process start to fork
I'm the Parents Progress, My pid = 5879
I'm the Child Progress, My pid = 5880
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
```

10. pipe

```
vagrant@csc3150:~/csc3150/template_source/source/program1$ ./program1 ./pipe
Process start to fork
I'm the Parents Progress, My pid = 17630
I'm the Child Progress, My pid = 17631
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
```

11. quit

```
vagrant@csc3150:~/csc3150/template_source/source/program1$ ./program1 ./quit
Process start to fork
I'm the Parents Progress, My pid = 17717
I'm the Child Progress, My pid = 17718
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
```

12. Segment_fault

```
vagrant@csc3150:~/csc3150/template_source/source/program1$ ./program1 ./segment_fault
Process start to fork
I'm the Parents Progress, My pid = 17771
I'm the Child Progress, My pid = 17772
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
```

13. stop

```
vagrant@csc3150:~/csc3150/template_source/source/program1$ ./program1 ./stop
Process start to fork
I'm the Parents Progress, My pid = 6053
I'm the Child Progress, My pid = 6054
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGSTOP program

Parent process receives SIGCHLD signal
```

14. Terminate

```
vagrant@csc3150:~/csc3150/template_source/source/program1$ ./program1 ./terminate
Process start to fork
I'm the Parents Progress, My pid = 17798
I'm the Child Progress, My pid = 17799
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
```

15. Trap

```
vagrant@csc3150:~/csc3150/template_source/source/program1$ ./program1 ./trap
Process start to fork
I'm the Parents Progress, My pid = 17825
I'm the Child Progress, My pid = 17826
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
```

2. Program2

2.1 Implementation

Kernel_clone is used to create the process. When the process is created, some arguments will be changed. We use do_exec to execute the program. In kernel mode, we use do_wait to wait for the signal. Kthread_creates a kernel to execute function. When using these four kernel functions, we need to find it in the kernel and export its symbol and the recompile the kernel. The main flow of task 2 is kernel module initializes, kernel thread creating, forking a process child process executing the test program, waiting until child process terminates and exiting the kernel module.

2.2 Output:

1. normal

```
[ 9526.178146] [program2] : module_init {Cong Yuzheng} {120090671}
[ 9526.178153] [program2] : module_init create kthread start
[ 9526.178259] [program2] : module_init kthread start
[ 9526.181771] [program2] : The child process has pid = 7566
[ 9526.181774] [program2] : This is the parent process, pid = 7564
[ 9526.181776] [program2] : child process
[ 9526.182731] [program2] : This is the normal
[ 9526.182735] [program2] : The return signal is 0
[ 9534.428414] [program2] : Module_exit
```

2. abort

```
[ 9712.126237] [program2] : module_init {Cong Yuzheng} {120090671}
[ 9712.126242] [program2] : module_init create kthread start
[ 9712.126370] [program2] : module_init kthread start
[ 9712.127527] [program2] : The child process has pid = 7797
[ 9712.127531] [program2] : This is the parent process, pid = 7795
[ 9712.127533] [program2] : child process
[ 9712.498587] [program2] : get SIGABRT signal
[ 9712.498616] [program2] : The return signal is 6
[ 9715.109024] [program2] : Module_exit
```

3. alarm

```
[ 9791.741444] [program2] : module_init {Cong Yuzheng} {120090671}
[ 9791.741448] [program2] : module_init create kthread start
[ 9791.741748] [program2] : module_init kthread start
[ 9791.741855] [program2] : The child process has pid = 7896
[ 9791.741857] [program2] : This is the parent process, pid = 7895
[ 9791.741859] [program2] : child process
[ 9791.742709] [program2] : get SIGALRM signal
[ 9791.742712] [program2] : The return signal is 14
[ 9794.950893] [program2] : Module_exit
```

4. bus

```
[ 9873.123031] [program2] : module_init {Cong Yuzheng} {120090671}
[ 9873.123036] [program2] : module_init create kthread start
[ 9873.123352] [program2] : module_init kthread start
[ 9873.125720] [program2] : The child process has pid = 7975
[ 9873.125723] [program2] : This is the parent process, pid = 7973
[ 9873.125724] [program2] : child process
[ 9873.489765] [program2] : get SIGBUS signal
[ 9873.489769] [program2] : The return signal is 7
[ 9877.889064] [program2] : Module_exit
```

5. floating

```
[ 9930.777397] [program2] : module_init {Cong Yuzheng} {120090671}
[ 9930.777402] [program2] : module_init create kthread start
[ 9930.777504] [program2] : module_init kthread start
[ 9930.781086] [program2] : The child process has pid = 8098
[ 9930.781089] [program2] : This is the parent process, pid = 8096
[ 9930.781091] [program2] : child process
[ 9931.158320] [program2] : get SIGFPE signal
[ 9931.158324] [program2] : The return signal is 8
[ 9934.786012] [program2] : Module_exit
```

6. hangup

```
[ 9995.634533] [program2] : module_init {Cong Yuzheng} {120090671}
[ 9995.634536] [program2] : module_init create kthread start
[ 9995.634598] [program2] : module_init kthread start
[ 9995.635180] [program2] : The child process has pid = 8156
[ 9995.635184] [program2] : This is the parent process, pid = 8155
[ 9995.635185] [program2] : child process
[ 9995.636228] [program2] : get SIGHUP signal
[ 9995.636232] [program2] : The return signal is 1
[ 9999.732006] [program2] : Module_exit
```

7. illegal_str

```
[10059.893693] [program2] : module_init {Cong Yuzheng} {120090671}
[10059.893698] [program2] : module_init create kthread start
[10059.893865] [program2] : module_init kthread start
[10059.896856] [program2] : The child process has pid = 8213
[10059.896860] [program2] : This is the parent process, pid = 8211
[10059.896862] [program2] : child process
[10060.252479] [program2] : get SIGILL signal
[10060.252484] [program2] : The return signal is 4
[10062.520584] [program2] : Module_exit
```

8. interrupt

```
[10176.761361] [program2] : module_init {Cong Yuzheng} {120090671}
[10176.761365] [program2] : module_init create kthread start
[10176.761666] [program2] : module_init kthread start
[10176.761766] [program2] : The child process has pid = 8313
[10176.761768] [program2] : This is the parent process, pid = 8312
[10176.761770] [program2] : child process
[10176.762898] [program2] : get SIGINT signal
[10176.762903] [program2] : The return signal is 2
[10181.242565] [program2] : Module_exit
```

9. kill

```
[10297.490065] [program2] : module_init {Cong Yuzheng} {120090671}
[10297.490070] [program2] : module_init create kthread start
[10297.490575] [program2] : module_init kthread start
[10297.490972] [program2] : The child process has pid = 8412
[10297.490977] [program2] : This is the parent process, pid = 8411
[10297.490980] [program2] : child process
[10297.492103] [program2] : get SIGKILL signal
[10297.492108] [program2] : The return signal is 9
[10301.443934] [program2] : Module_exit
```

10. pipe

```
[10371.173803] [program2] : module_init {Cong Yuzheng} {120090671}
[10371.173808] [program2] : module_init create kthread start
[10371.174087] [program2] : module_init kthread start
[10371.177369] [program2] : The child process has pid = 8471
[10371.177373] [program2] : This is the parent process, pid = 8469
[10371.177375] [program2] : child process
[10371.178623] [program2] : get SIGPIPE signal
[10371.178628] [program2] : The return signal is 13
[10374.722865] [program2] : Module_exit
```

11. quit

```
[10436.169783] [program2] : module_init {Cong Yuzheng} {120090671}
[10436.169787] [program2] : module_init create kthread start
[10436.170062] [program2] : module_init kthread start
[10436.170163] [program2] : The child process has pid = 8611
[10436.170166] [program2] : This is the parent process, pid = 8610
[10436.170168] [program2] : child process
[10436.526768] [program2] : get SIGQUIT signal
[10436.526771] [program2] : The return signal is 3
[10439.645495] [program2] : Module_exit
```

12.Segment_fault

```
[10520.748434] [program2] : module_init {Cong Yuzheng} {120090671}
[10520.748441] [program2] : module_init create kthread start
[10520.748549] [program2] : module_init kthread start
[10520.752377] [program2] : The child process has pid = 8675
[10520.752380] [program2] : This is the parent process, pid = 8673
[10520.752382] [program2] : child process
[10521.109233] [program2] : get SIGSEGV signal
[10521.109236] [program2] : The return signal is 11
[10524.069597] [program2] : Module_exit
```

13. Stop

```
[12212.425839] [program2] : module_init {Cong Yuzheng} {120090671}
[12212.425846] [program2] : module_init create kthread start
[12212.425967] [program2] : module_init kthread start
[12212.427215] [program2] : The child process has pid = 14250
[12212.427219] [program2] : This is the parent process, pid = 14248
[12212.427220] [program2] : child process
[12212.429970] [program2] : get SIGSTOP signal
[12212.429977] [program2] : The return signal is 19
[12216.049893] [program2] : Module_exit
```

14.Terminate

```
[10987.367437] [program2] : module_init {Cong Yuzheng} {120090671}
[10987.367441] [program2] : module_init create kthread start
[10987.367692] [program2] : module_init kthread start
[10987.367817] [program2] : The child process has pid = 8894
[10987.367820] [program2] : This is the parent process, pid = 8893
[10987.367822] [program2] : child process
[10987.369081] [program2] : get SIGTERM signal
[10987.369084] [program2] : The return signal is 15
[10990.740157] [program2] : Module_exit
```

15.Trap

```
[11109.848656] [program2] : module_init {Cong Yuzheng} {120090671}
[11109.848661] [program2] : module_init create kthread start
[11109.848930] [program2] : module_init kthread start
[11109.849030] [program2] : The child process has pid = 8976
[11109.849033] [program2] : This is the parent process, pid = 8975
[11109.849035] [program2] : child process
[11110.224582] [program2] : get SIGTRAP signal
[11110.224586] [program2] : The return signal is 5
[11113.246429] [program2] : Module_exit
```

3. Development environment set up.

3.1 Download the source code from kernel.org. and extract it.

3.2 Clean the previous settings by make prproper, make clean make menuconfig

3.3 Build kernel Image and modules by make bzImage -j$(nproc), make modules -j$(nproc), make -j$(nproc)

3.4 Install kernel modules and install kernel.

This is the first time of installing, after that we need to add some export

signal to the kernel and recompile it in order to use kernel functions.

4. Learning outcomes:

4.1 Better understanding of what the kernel mode did when forking a process

4.2 know how to use the clang-format

4.3 Better understanding of C knowledge.

4.4 Be able to try to read some kernel codes of Linux

4.5 know how to compile the kernel