CSC3150_Assignment1_Report

Liu Yuheng 120090263

1.Design

In this program, I need to write two programs for user mode implementation and for kernel mode implementation.

Program1

1) General idea

The process for user mode is to first fork a child process to execute fifteen test programs, and the parent process will receive the SIGCHLD signals when the child process finishes execution.

I distinguished different SIGCHLD including "SIGHUP", "SIGINT", "SIGQUIT", "SIGILL", "SIGTRAP", "SIGABRT", "SIGBUS", "SIGFPE", "SIGKILL", "SIGSEGV", "SIGPIPE", "SIGALRM", "SIGTERM", "SIGSTOP" so that the program can print corresponding signals due to different return status.

2) Detailed information

- In my program, I first defined different termination signals (char* TerminationSignal[ ]). Then, there is a main() function.
- First, I call fork() to create a child process. The fork call is called only once but returns twice. In the parent process, it returns the process ID of the child process. In the child process, it returns 0. Therefore, if successfully fork child process, the program first gets into child process. I use execve() function to execute test programs and use exit() function to exit.
- During this process, the parent process is halted and waiting for the child process by waitpid() fucntions. Then, I check different return status and print out corresponding signals. At the end of the parent process, I use exit(0) to exit it.

Steps to test:

- make
- use ./program1 ./xxxx to test

Program2

1) General idea

The general idea of program2 is a little similar to program1.

The main process is also to fork a child process and the parent process will wait until child process terminates. However, this program is implemented in kernel mode.

Therefore, I have to first initialize kernel modules, create a kernel thread and then fork a process. After the parent process wait until the child process terminates, the kernel module exits.

2) Detailed information

Preparations:

- I export functions including kernel_theard() from '/kernel/fork.c', getname_kernel() from '/fs/namei.c', do_wait() from '/kernel/exit.c', do_execve() from '/fs/exec.c' by EXPORT_SYMBOL. I copy the wait_opts{} structure of Linux 15.10.99 from '/kernel/exit.c' to my program2. I declare task_struct structure. I copy the definition of functions including WEXITSTATUS, WTERMSIG, WSTOPSIG, WIFEXITED, WIFSIGNALED, WIFSTOPPED from '/x86_64-linux-gnu/sys/wait.h' file.

Process:

- My program includes function my_wait(), my_exec(), my_fork(), and program2_init(), progarm2_exit().
- In my_wait(), I called find_get_pid(), wait() and put_pid() in sequence, then I print termination information accordingly.
- In my_exec(), I set path (/tmp/testfor submission and /home/vagrant/csc3150/Assignment_1_120090263/xxxxx for test), argument vector, environment. I use getname_kernel() to obtain filename and use do_execve() to execute the test file. If the return value of do_execve() is 1 instead of 0, then the result has errors, otherwise the result is correct . Lastly, I use do_exit(result) as the process finishes running or exits because a system exception is triggered
- In my_fork(), first set default sigaction for current process. Then I use kernel_clone() to fork a process. However, since there are some problems caused by kernel_clone(), including the wrong return PID, I changed the parameters to that of kernel_thread() and solve this problem. In this function, I called my_wait().
- program2_init() is for creating a kernel thread to run my_fork(), which will use kthread_create() and use IS_ERR() to judge whether the process is successful. If so, it use wake_up_process() to wake up new thread.
- progarm2_exit() marks the end of whole program2.

Steps to test:

- use gcc test.c -o test to compile test files
- change the path to corresponding files
- make
- insert and remove module: insmod program2.ko, rmmod program2.o
- use dmesg to check output.

2. Set development environment

- Linux version:5.10.99
- Ubuntu version:16.04.12
- Virtual Box 6.1

steps:

1. set up virtual machine
    1. install virtualbox and vagrant
    2. make a directory for csc3150
    3. Launch powershell with Administrator previlege and change current directory to execute 'vagrant init cyzhu/csc3150'
    4. execute vagrant up
    5. execute vagrant ssh
2. set up VS Code
    1. download Remote SSH
    2. go to the remote explorer tab, click config in SSH-TARGETS
    3. go back to powershell and execute vagrant ssh-config
    4. find SSH Target called default and click the icon to connect to the VM and launch a new window.
    5. install essential dependencies and libraries: sudo apt update && sudo apt install -y build-essential
    6. create a directory for the course: mkdir -p ~/csc3150

Compile kernel

- Download: I download source code by wget method. (Linux 5.10.99)
- Preparation: I extract the source file to /home/vagrant/. Then I copy config file from /boot to /home/vagrant/Linux 5.10.99. I install development tools.
- Process: I follow the instructions in the order of make mrproper, make clean, make menuconfig, make bzImage, make modules, make modules_install, make install and reboot to switch to the required kernel version.

3.What I learned from this project

- From program1, I learn how to create child process and execute other programs. I get to know the basic knowledge of how programs in user mode are executed.
- From program2, I learn how to modify kernel files, change kernel versions and compile kernel. Also, I get to know what termination signals are and what situation they are responsible for. In this process, I also enhanced my skills of using Linux instrutions.

4. Test program screen shots

Program1

Program1_abort

```
root@csc3150:/home/vagrant/csc3150/source/program1# ./program1 ./abort
Process starts to fork
I'm the FATHER Process: pid = 27452
I'm the CHILD Process: pid = 27453
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGABRT program

This is termination status:
Parent process receives  SIGABRT signal
CHILD Execution Failed: 6
root@csc3150:/home/vagrant/csc3150/source/program1#
```

Program1_alarm

```
root@csc3150:/home/vagrant/csc3150/source/program1# ./program1 ./alarm
Process starts to fork
I'm the FATHER Process: pid = 27519
I'm the CHILD Process: pid = 27520
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGALRM program

This is termination status:
Parent process receives  SIGALRM signal
CHILD Execution Failed: 14
root@csc3150:/home/vagrant/csc3150/source/program1#
```

Program1_bus

```
root@csc3150:/home/vagrant/csc3150/source/program1# ./program1 ./bus
Process starts to fork
I'm the FATHER Process: pid = 27622
I'm the CHILD Process: pid = 27623
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGBUS program

This is termination status:
Parent process receives  SIGBUS signal
CHILD Execution Failed: 7
root@csc3150:/home/vagrant/csc3150/source/program1#
```

Program1_floating

```
root@csc3150:/home/vagrant/csc3150/source/program1# ./program1 ./floating
Process starts to fork
I'm the FATHER Process: pid = 27667
I'm the CHILD Process: pid = 27668
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGFPE program

This is termination status:
Parent process receives  SIGFPE signal
CHILD Execution Failed: 8
root@csc3150:/home/vagrant/csc3150/source/program1#
```

Program1_hangup

```
root@csc3150:/home/vagrant/csc3150/source/program1# ./program1 ./hangup
Process starts to fork
I'm the FATHER Process: pid = 27700
I'm the CHILD Process: pid = 27701
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGHUP program

This is termination status:
Parent process receives  SIGHUP signal
CHILD Execution Failed: 1
root@csc3150:/home/vagrant/csc3150/source/program1#
```

Program1_illegal_instr

```
root@csc3150:/home/vagrant/csc3150/source/program1# ./program1 ./illegal_instr
Process starts to fork
I'm the FATHER Process: pid = 27735
I'm the CHILD Process: pid = 27736
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGILL program

This is termination status:
Parent process receives  SIGILL signal
CHILD Execution Failed: 4
root@csc3150:/home/vagrant/csc3150/source/program1#
```

Program1_interrupt

```
root@csc3150:/home/vagrant/csc3150/source/program1# ./program1 ./interrupt
Process starts to fork
I'm the FATHER Process: pid = 27789
I'm the CHILD Process: pid = 27790
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGINT program

This is termination status:
Parent process receives  SIGINT signal
CHILD Execution Failed: 2
root@csc3150:/home/vagrant/csc3150/source/program1#
```

Program1_kill

```
root@csc3150:/home/vagrant/csc3150/source/program1# ./program1 ./kill
Process starts to fork
I'm the FATHER Process: pid = 27833
I'm the CHILD Process: pid = 27834
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGKILL program

This is termination status:
Parent process receives  SIGKILL signal
CHILD Execution Failed: 9
root@csc3150:/home/vagrant/csc3150/source/program1#
```

Program1_normal

```
root@csc3150:/home/vagrant/csc3150/source/program1# ./program1 ./normal
Process starts to fork
I'm the FATHER Process: pid = 27899
I'm the CHILD Process: pid = 27900
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the normal program

------------CHILD PROCESS END------------
This is termination status:
Parent process receives SIGCHLD signal
Normal termination with Exit status: 0
root@csc3150:/home/vagrant/csc3150/source/program1#
```

Program1_pipe

```
root@csc3150:/home/vagrant/csc3150/source/program1# ./program1 ./pipe
Process starts to fork
I'm the FATHER Process: pid = 27944
I'm the CHILD Process: pid = 27945
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGPIPE program

This is termination status:
Parent process receives  SIGPIPE signal
CHILD Execution Failed: 13
root@csc3150:/home/vagrant/csc3150/source/program1#
```

Program1_quit

```
root@csc3150:/home/vagrant/csc3150/source/program1# ./program1 ./quit
Process starts to fork
I'm the CHILD Process: pid = 27989
I'm the FATHER Process: pid = 27988
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGQUIT program

This is termination status:
Parent process receives  SIGQUIT signal
CHILD Execution Failed: 3
root@csc3150:/home/vagrant/csc3150/source/program1#
```

Program1_segment_fault

```
root@csc3150:/home/vagrant/csc3150/source/program1# ./program1 ./segment_fault
Process starts to fork
I'm the FATHER Process: pid = 28054
I'm the CHILD Process: pid = 28055
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGSEGV program

This is termination status:
Parent process receives  SIGSEGV signal
CHILD Execution Failed: 11
root@csc3150:/home/vagrant/csc3150/source/program1#
```

Program1_stop

```
root@csc3150:/home/vagrant/csc3150/source/program1# ./program1 ./stop
Process starts to fork
I'm the FATHER Process: pid = 28099
I'm the CHILD Process: pid = 28100
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGSTOP program

This is termination status:
Parent process receives SIGSTOP signal , status: 19
CHILD PROCESS STOPPED
root@csc3150:/home/vagrant/csc3150/source/program1#
```

Program1_terminate

```
root@csc3150:/home/vagrant/csc3150/source/program1# ./program1 ./terminate
Process starts to fork
I'm the FATHER Process: pid = 28143
I'm the CHILD Process: pid = 28144
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGTERM program

This is termination status:
Parent process receives  SIGTERM signal
CHILD Execution Failed: 15
root@csc3150:/home/vagrant/csc3150/source/program1#
```

Program1_trap

```
root@csc3150:/home/vagrant/csc3150/source/program1# ./program1 ./trap
Process starts to fork
I'm the FATHER Process: pid = 28187
I'm the CHILD Process: pid = 28188
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGTRAP program

This is termination status:
Parent process receives  SIGTRAP signal
CHILD Execution Failed: 5
root@csc3150:/home/vagrant/csc3150/source/program1#
```

Program2

Program2_abort

```
[ 3730.333804] [program2] : Module_init {Liu_Yuheng} {ID:120090263}
[ 3730.353048] [program2] : Module_init create kthread start
[ 3730.368977] [program2] : Module_init kthread start
[ 3730.384096] [program2] : The child process has pid = 26995
[ 3730.384097] [program2] : This is the parent process, pid =  26994
[ 3730.406254] [program2] : child process
[ 3730.512259] [program2] : get SIGABRT signal
[ 3730.526283] [program2] : child process terminated
[ 3730.526283] [program2] : The return signal is 6
[ 3732.281838] [program2] : Module_exit
root@csc3150:/home/vagrant/csc3150/source/program2#
```

Program2_alarm

```
[ 3697.375624] [program2] : Module_init {Liu_Yuheng} {ID:120090263}
[ 3697.396692] [program2] : Module_init create kthread start
[ 3697.414744] [program2] : Module_init kthread start
[ 3697.431431] [program2] : The child process has pid = 26572
[ 3697.431432] [program2] : This is the parent process, pid =  26571
[ 3697.449365] [program2] : child process
[ 3699.435871] [program2] : get SIGALRM signal
[ 3699.450632] [program2] : child process terminated
[ 3699.450634] [program2] : The return signal is 14
[ 3700.500007] [program2] : Module_exit
root@csc3150:/home/vagrant/csc3150/source/program2#
```

Program2_bus

```
[ 3633.477809] [program2] : Module_init {Liu_Yuheng} {ID:120090263}
[ 3633.502823] [program2] : Module_init create kthread start
[ 3633.519466] [program2] : Module_init kthread start
[ 3633.536982] [program2] : The child process has pid = 25110
[ 3633.536983] [program2] : This is the parent process, pid =  25109
[ 3633.557007] [program2] : child process
[ 3633.640290] [program2] : get SIGBUS signal
[ 3633.654890] [program2] : child process terminated
[ 3633.654890] [program2] : The return signal is 7
[ 3634.756370] [program2] : Module_exit
root@csc3150:/home/vagrant/csc3150/source/program2#
```

Program2_floating

```
[ 3598.998295] [program2] : Module_init {Liu_Yuheng} {ID:120090263}
[ 3599.029318] [program2] : Module_init create kthread start
[ 3599.048861] [program2] : Module_init kthread start
[ 3599.065304] [program2] : The child process has pid = 24686
[ 3599.065305] [program2] : This is the parent process, pid =  24685
[ 3599.083153] [program2] : child process
[ 3599.169308] [program2] : get SIGFPE signal
[ 3599.182974] [program2] : child process terminated
[ 3599.182974] [program2] : The return signal is 8
[ 3600.309062] [program2] : Module_exit
root@csc3150:/home/vagrant/csc3150/source/program2#
```

Program2_hangup

```
[ 3564.408047] [program2] : Module_init {Liu_Yuheng} {ID:120090263}
[ 3564.428849] [program2] : Module_init create kthread start
[ 3564.448351] [program2] : Module_init kthread start
[ 3564.470274] [program2] : The child process has pid = 24202
[ 3564.470275] [program2] : This is the parent process, pid =  24201
[ 3564.487565] [program2] : child process
[ 3564.520880] [program2] : get SIGHUP signal
[ 3564.536837] [program2] : child process terminated
[ 3564.536838] [program2] : The return signal is 1
[ 3566.025550] [program2] : Module_exit
root@csc3150:/home/vagrant/csc3150/source/program2#
```

Program2_illegal_instr

```
[ 3533.939115] [program2] : Module_init {Liu_Yuheng} {ID:120090263}
[ 3533.954895] [program2] : Module_init create kthread start
[ 3533.968833] [program2] : Module_init kthread start
[ 3533.982070] [program2] : The child process has pid = 23777
[ 3533.982071] [program2] : This is the parent process, pid =  23776
[ 3534.000973] [program2] : child process
[ 3534.095479] [program2] : get SIGILL signal
[ 3534.113140] [program2] : child process terminated
[ 3534.113141] [program2] : The return signal is 4
[ 3536.010904] [program2] : Module_exit
root@csc3150:/home/vagrant/csc3150/source/program2#
```

Program2_interrupt

```
[ 3501.929548] [program2] : Module_init {Liu_Yuheng} {ID:120090263}
[ 3501.951418] [program2] : Module_init create kthread start
[ 3501.981211] [program2] : Module_init kthread start
[ 3501.998033] [program2] : The child process has pid = 23343
[ 3501.998034] [program2] : This is the parent process, pid =  23342
[ 3502.016748] [program2] : child process
[ 3502.050224] [program2] : get SIGINT signal
[ 3502.064760] [program2] : child process terminated
[ 3502.064761] [program2] : The return signal is 2
[ 3503.364212] [program2] : Module_exit
root@csc3150:/home/vagrant/csc3150/source/program2#
```

Program2_kill

```
[ 3460.493540] [program2] : Module_init {Liu_Yuheng} {ID:120090263}
[ 3460.510031] [program2] : Module_init create kthread start
[ 3460.524921] [program2] : Module_init kthread start
[ 3460.537448] [program2] : The child process has pid = 22908
[ 3460.537448] [program2] : This is the parent process, pid =  22907
[ 3460.556322] [program2] : child process
[ 3460.629011] [program2] : get SIGKILL signal
[ 3460.644691] [program2] : child process terminated
[ 3460.644691] [program2] : The return signal is 9
[ 3462.411408] [program2] : Module_exit
root@csc3150:/home/vagrant/csc3150/source/program2#
```

Program2_normal

```
[ 3356.243653] [program2] : Module_init {Liu_Yuheng} {ID:120090263}
[ 3356.262733] [program2] : Module_init create kthread start
[ 3356.280807] [program2] : Module_init kthread start
[ 3356.297859] [program2] : The child process has pid = 22070
[ 3356.297860] [program2] : This is the parent process, pid =  22069
[ 3356.314634] [program2] : child process
[ 3356.340462] [program2] : Child process gets NORMAL termination and return signal is 0
[ 3357.548395] [program2] : Module_exit
root@csc3150:/home/vagrant/csc3150/source/program2#
```

Program2_pipe

```
[ 3329.554116] [program2] : Module_init {Liu_Yuheng} {ID:120090263}
[ 3329.573988] [program2] : Module_init create kthread start
[ 3329.592739] [program2] : Module_init kthread start
[ 3329.609456] [program2] : The child process has pid = 21655
[ 3329.609459] [program2] : This is the parent process, pid =  21654
[ 3329.624188] [program2] : child process
[ 3329.651663] [program2] : get SIGPIPE signal
[ 3329.663217] [program2] : child process terminated
[ 3329.663218] [program2] : The return signal is 13
[ 3330.885903] [program2] : Module_exit
root@csc3150:/home/vagrant/csc3150/source/program2#
```

Program2_quit

```
[ 3302.439470] [program2] : Module_init {Liu_Yuheng} {ID:120090263}
[ 3302.455261] [program2] : Module_init create kthread start
[ 3302.469533] [program2] : Module_init kthread start
[ 3302.482401] [program2] : The child process has pid = 21231
[ 3302.482402] [program2] : This is the parent process, pid =  21230
[ 3302.499530] [program2] : child process
[ 3302.579751] [program2] : get SIGQUIT signal
[ 3302.593710] [program2] : child process terminated
[ 3302.593711] [program2] : The return signal is 3
[ 3303.896228] [program2] : Module_exit
root@csc3150:/home/vagrant/csc3150/source/program2#
```

Program2_segment_fault

```
[ 3263.950909] [program2] : Module_init {Liu_Yuheng} {ID:120090263}
[ 3263.978063] [program2] : Module_init create kthread start
[ 3263.996997] [program2] : Module_init kthread start
[ 3264.014720] [program2] : The child process has pid = 20806
[ 3264.014721] [program2] : This is the parent process, pid =  20805
[ 3264.031530] [program2] : child process
[ 3264.109680] [program2] : get SIGSEGV signal
[ 3264.122992] [program2] : child process terminated
[ 3264.122993] [program2] : The return signal is 11
[ 3264.868105] [program2] : Module_exit
root@csc3150:/home/vagrant/csc3150/source/program2#
```

Program2_stop

```
[ 3229.464654] [program2] : Module_init {Liu_Yuheng} {ID:120090263}
[ 3229.480136] [program2] : Module_init create kthread start
[ 3229.494620] [program2] : Module_init kthread start
[ 3229.507334] [program2] : The child process has pid = 20418
[ 3229.507335] [program2] : This is the parent process, pid =  20417
[ 3229.526345] [program2] : child process
[ 3229.567934] [program2] : get SIGSTOP signal
[ 3229.581891] [program2] : child process terminated
[ 3229.581892] [program2] : The return signal is 19
[ 3230.838845] [program2] : Module_exit
root@csc3150:/home/vagrant/csc3150/source/program2#
```

Program2_terminate

```
[ 3097.788932] [program2] : Module_init {Liu_Yuheng} {ID:120090263}
[ 3097.813131] [program2] : Module_init create kthread start
[ 3097.831185] [program2] : Module_init kthread start
[ 3097.844500] [program2] : The child process has pid = 19938
[ 3097.844500] [program2] : This is the parent process, pid =  19936
[ 3097.859144] [program2] : child process
[ 3097.887023] [program2] : get SIGTERM signal
[ 3097.898237] [program2] : child process terminated
[ 3097.898238] [program2] : The return signal is 15
[ 3099.250484] [program2] : Module_exit
root@csc3150:/home/vagrant/csc3150/source/program2#
```

Program2_trap

```
[ 3010.158797] [program2] : Module_init {Liu_Yuheng} {ID:120090263}
[ 3010.176028] [program2] : Module_init create kthread start
[ 3010.192218] [program2] : Module_init kthread start
[ 3010.205409] [program2] : The child process has pid = 19509
[ 3010.205410] [program2] : This is the parent process, pid =  19508
[ 3010.223835] [program2] : child process
[ 3010.313281] [program2] : get SIGTRAP signal
[ 3010.329137] [program2] : child process terminated
[ 3010.329138] [program2] : The return signal is 5
[ 3011.791629] [program2] : Module_exit
root@csc3150:/home/vagrant/csc3150/source/program2#
```