Report

Shaoqiang Sun 120090638

How did you design your program?

Program1:

main() calls fork() to fork a process.

The child process prints out its pid and calls execve() to execute the test program.

The parent process prints out its pid and calls waitpid() to receive the SIGCHLD signal when child process finishes execution.

The parent process calls WIFEXITED() WIFSIGNALED() WIFSTOPPED() to check the status of the child process.

The parent process calls WEXITSTATUS() WTERMSIG() WSTOPSIG() to print out the termination information of child process according to the signal received.

Program2:

When program2.ko is initialized, program2_init() calls kthread_create() to create a kennel thread and calls my_fork().

my_fork() sets default sigaction for the current process, and calls kernel_clone() to fork a process.

The pid of the child process and the parent process will be printed out.

The child process executes my_exec().

my_exec() calls getname_kernel() to get the test program file name and calls do_execve() to execute the test program.

The test program will raise signal.

The parent process calls my_wait().

my_wait() calls do_wait() to wait until the child process terminates, and catches the signal raised in the child process.

The parent process prints out the termination information of child process according to the signal received.

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

Complie the kernel:

Download the mirror kernel source code

Enter *wget* *https://mirror.tuna.tsinghua.edu.cn/kernel/v5.x/linux-5.10.5.tar.gz* in the terminal.

Install Dependency and development tools

Enter *sudo apt-get install libncurses-dev gawk flex bison openssl libssl-dev dkms libelf-dev libudev- dev libpci-dev libiberty-dev autoconf llvm dwarves* in the terminal.

Extract the source file to /home/seed/work/

Copy config from /boot to /home/seed/work/ linux-5.10.5

Login root account and go to /home/seed/work/ linux-5.10.5

Start configuration

Enter *make menuconfig* in the terminal, load config and save config

Build kernel Image and module

Enter *make bzImage -j(nproc), make modules -j$(nproc)* in the terminal

Install kernel module

Enter *$make modules_install* in the terminal

Install kernel

Enter *make install* in the terminal

Reboot to load new kernel

Enter *reboot* in the terminal

Export symbol:

EXPORT_SYMBOL() under the definition of kernel_clone() do_execve() getname_kernel() do_wait() in the kernel source code

Extern kernel_clone() do_execve() getname_kernel() do_wait() in program2.c

Recompile the kernel:

Build kernel Image and module

Enter *make bzImage -j(nproc), make modules -j$(nproc)* in the terminal

Install kernel module

Enter *$make modules_install* in the terminal

Install kernel

Enter *make install* in the terminal

Reboot to load new kernel

Enter *reboot* in the terminal

Screenshot of your program output.

Program1:

```
vagrant@csc3150:~/csc3150/Assignment_1_120090638/source/program1$ ./program1 ./normal
Process start to fork
I'm the Parent Process, my pid = 5554
I'm the Child Process, my pid = 5555
Child process starts 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/Assignment_1_120090638/source/program1$ ./program1 ./abort
Process start to fork
I'm the Parent Process, my pid = 5590
I'm the Child Process, my pid = 5591
Child process starts 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/Assignment_1_120090638/source/program1$ ./program1 ./alarm
Process start to fork
I'm the Parent Process, my pid = 5799
I'm the Child Process, my pid = 5800
Child process starts to execute test program:
------------CHILD PROCESS START------------
This is the SIGALRM program

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

```
vagrant@csc3150:~/csc3150/Assignment_1_120090638/source/program1$ ./program1 ./bus
Process start to fork
I'm the Parent Process, my pid = 5874
I'm the Child Process, my pid = 5875
Child process starts 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/Assignment_1_120090638/source/program1$ ./program1 ./floating
Process start to fork
I'm the Parent Process, my pid = 5903
I'm the Child Process, my pid = 5904
Child process starts 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/Assignment_1_120090638/source/program1$ ./program1 ./hangup
Process start to fork
I'm the Parent Process, my pid = 5918
I'm the Child Process, my pid = 5919
Child process starts 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/Assignment_1_120090638/source/program1$ ./program1 ./illegal_instr
Process start to fork
I'm the Parent Process, my pid = 6000
I'm the Child Process, my pid = 6001
Child process starts 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/Assignment_1_120090638/source/program1$ ./program1 ./interrupt
Process start to fork
I'm the Parent Process, my pid = 6040
I'm the Child Process, my pid = 6041
Child process starts 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/Assignment_1_120090638/source/program1$ ./program1 ./kill
Process start to fork
I'm the Parent Process, my pid = 6099
I'm the Child Process, my pid = 6100
Child process starts 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/Assignment_1_120090638/source/program1$ ./program1 ./pipe
Process start to fork
I'm the Parent Process, my pid = 6168
I'm the Child Process, my pid = 6169
Child process starts 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/Assignment_1_120090638/source/program1$ ./program1 ./quit
Process start to fork
I'm the Parent Process, my pid = 6202
I'm the Child Process, my pid = 6203
Child process starts 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/Assignment_1_120090638/source/program1$ ./program1 ./segment_fault
Process start to fork
I'm the Parent Process, my pid = 6247
I'm the Child Process, my pid = 6248
Child process starts 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/Assignment_1_120090638/source/program1$ ./program1 ./terminate
Process start to fork
I'm the Parent Process, my pid = 6332
I'm the Child Process, my pid = 6333
Child process starts 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/Assignment_1_120090638/source/program1$ ./program1 ./trap
Process start to fork
I'm the Parent Process, my pid = 6365
I'm the Child Process, my pid = 6366
Child process starts to execute test program:
------------CHILD PROCESS START------------
This is the SIGTRAP program

Parent process receives SIGCHLD signal
child process get SIGTRAP signal
```

```
vagrant@csc3150:~/csc3150/Assignment_1_120090638/source/program1$ ./program1 ./stop
Process start to fork
I'm the Parent Process, my pid = 6275
I'm the Child Process, my pid = 6276
Child process starts to execute test program:
------------CHILD PROCESS START------------
This is the SIGSTOP program

Parent process receives SIGCHLD signal
child process get SIGSTOP signal
```

Program2:

```
[  206.845838] [program2] : module_init {Shaoqiang Sun} {120090638}
[  206.845841] [program2] : module_init create kthread start
[  206.845980] [program2] : module_init kthread start
[  206.846078] [program2] : The child process has pid = 3270
[  206.846079] [program2] : This is the parent process, pid = 3269
[  207.171639] [program2] : child process
[  207.171641] [program2] : get SIGABRT signal
[  207.171641] [program2] : child process Abort (ANSI).
[  207.171642] [program2] : The return signal is 6
[  209.187867] [program2] : module_exit
```

```
[  257.976109] [program2] : module_init {Shaoqiang Sun} {120090638}
[  257.976112] [program2] : module_init create kthread start
[  257.976335] [program2] : module_init kthread start
[  257.976420] [program2] : The child process has pid = 3301
[  257.976422] [program2] : This is the parent process, pid = 3300
[  259.982116] [program2] : child process
[  259.982118] [program2] : get SIGALRM signal
[  259.982118] [program2] : child process Alarm clock (POSIX).
[  259.982120] [program2] : The return signal is 14
[  262.827515] [program2] : module_exit
```

```
[  350.547447] [program2] : module_init {Shaoqiang Sun} {120090638}
[  350.547449] [program2] : module_init create kthread start
[  350.547664] [program2] : module_init kthread start
[  350.547810] [program2] : The child process has pid = 3355
[  350.547812] [program2] : This is the parent process, pid = 3354
[  350.736888] [program2] : child process
[  350.736890] [program2] : get SIGBUS signal
[  350.736891] [program2] : child process BUS error (4.2 BSD).
[  350.736892] [program2] : The return signal is 7
[  352.210291] [program2] : module_exit
```

```
[  545.576630] [program2] : module_init {Shaoqiang Sun} {120090638}
[  545.576633] [program2] : module_init create kthread start
[  545.576902] [program2] : module_init kthread start
[  545.576998] [program2] : The child process has pid = 3506
[  545.576999] [program2] : This is the parent process, pid = 3505
[  545.775629] [program2] : child process
[  545.775631] [program2] : get SIGFPE signal
[  545.775631] [program2] : child process Floating-point exception (ANSI).
[  545.775632] [program2] : The return signal is 8
[  548.274143] [program2] : module_exit
```

```
[  595.739585] [program2] : module_init {Shaoqiang Sun} {120090638}
[  595.739587] [program2] : module_init create kthread start
[  595.739725] [program2] : module_init kthread start
[  595.739836] [program2] : The child process has pid = 3571
[  595.739837] [program2] : This is the parent process, pid = 3570
[  595.740559] [program2] : child process
[  595.740561] [program2] : get SIGHUP signal
[  595.740561] [program2] : child process Hangup (POSIX).
[  595.740562] [program2] : The return signal is 1
[  597.038122] [program2] : module_exit
```

```
[  653.317816] [program2] : module_init {Shaoqiang Sun} {120090638}
[  653.317818] [program2] : module_init create kthread start
[  653.317968] [program2] : module_init kthread start
[  653.318035] [program2] : The child process has pid = 3633
[  653.318037] [program2] : This is the parent process, pid = 3632
[  653.515946] [program2] : child process
[  653.515948] [program2] : get SIGILL signal
[  653.515948] [program2] : child process Illegal instruction (ANSI).
[  653.515949] [program2] : The return signal is 4
[  654.764204] [program2] : module_exit
```

```
[  774.827354] [program2] : module_init {Shaoqiang Sun} {120090638}
[  774.827356] [program2] : module_init create kthread start
[  774.827444] [program2] : module_init kthread start
[  774.827481] [program2] : The child process has pid = 3701
[  774.827483] [program2] : This is the parent process, pid = 3700
[  774.828292] [program2] : child process
[  774.828293] [program2] : get SIGINT signal
[  774.828294] [program2] : child process Interrupt (ANSI).
[  774.828295] [program2] : The return signal is 2
[  776.377160] [program2] : module_exit
```

```
[  820.440850] [program2] : module_init {Shaoqiang Sun} {120090638}
[  820.440852] [program2] : module_init create kthread start
[  820.441035] [program2] : module_init kthread start
[  820.441146] [program2] : The child process has pid = 3754
[  820.441147] [program2] : This is the parent process, pid = 3753
[  820.441663] [program2] : child process
[  820.441664] [program2] : get SIGKILL signal
[  820.441665] [program2] : child process Kill, unblockable (POSIX).
[  820.441666] [program2] : The return signal is 9
[  821.944194] [program2] : module_exit
```

```
[  859.146995] [program2] : module_init {Shaoqiang Sun} {120090638}
[  859.146998] [program2] : module_init create kthread start
[  859.147096] [program2] : module_init kthread start
[  859.147207] [program2] : The child process has pid = 3793
[  859.147209] [program2] : This is the parent process, pid = 3792
[  859.147770] [program2] : child process
[  859.147771] [program2] : get SIGPIPE signal
[  859.147772] [program2] : child process Broken pipe (POSIX).
[  859.147772] [program2] : The return signal is 13
[  860.561068] [program2] : module_exit
```

```
[  898.185426] [program2] : module_init {Shaoqiang Sun} {120090638}
[  898.185428] [program2] : module_init create kthread start
[  898.185628] [program2] : module_init kthread start
[  898.185758] [program2] : The child process has pid = 3832
[  898.185759] [program2] : This is the parent process, pid = 3831
[  898.390681] [program2] : child process
[  898.390682] [program2] : get SIGQUIT signal
[  898.390683] [program2] : child process Quit (POSIX).
[  898.390684] [program2] : The return signal is 3
[  899.376337] [program2] : module_exit
```

```
[  944.047026] [program2] : module_init {Shaoqiang Sun} {120090638}
[  944.047028] [program2] : module_init create kthread start
[  944.047129] [program2] : module_init kthread start
[  944.047201] [program2] : The child process has pid = 3885
[  944.047203] [program2] : This is the parent process, pid = 3884
[  944.247517] [program2] : child process
[  944.247518] [program2] : get SIGSEGV signal
[  944.247519] [program2] : child process Segmentation violation (ANSI).
[  944.247520] [program2] : The return signal is 11
[  945.490982] [program2] : module_exit
```

```
[ 1042.626905] [program2] : module_init {Shaoqiang Sun} {120090638}
[ 1042.626923] [program2] : module_init create kthread start
[ 1042.627140] [program2] : module_init kthread start
[ 1042.627726] [program2] : The child process has pid = 3983
[ 1042.627728] [program2] : This is the parent process, pid = 3981
[ 1042.628493] [program2] : child process
[ 1042.628495] [program2] : get SIGTERM signal
[ 1042.628495] [program2] : child process Termination (ANSI).
[ 1042.628496] [program2] : The return signal is 15
[ 1043.859211] [program2] : module_exit
```

```
[ 1095.097026] [program2] : module_init {Shaoqiang Sun} {120090638}
[ 1095.097028] [program2] : module_init create kthread start
[ 1095.097270] [program2] : module_init kthread start
[ 1095.097395] [program2] : The child process has pid = 4034
[ 1095.097397] [program2] : This is the parent process, pid = 4033
[ 1095.284684] [program2] : child process
[ 1095.284685] [program2] : get SIGTRAP signal
[ 1095.284686] [program2] : child process Trace trap (POSIX).
[ 1095.284687] [program2] : The return signal is 5
[ 1096.366391] [program2] : module_exit
```

```
[ 1005.936895] [program2] : module_init {Shaoqiang Sun} {120090638}
[ 1005.936897] [program2] : module_init create kthread start
[ 1005.937209] [program2] : module_init kthread start
[ 1005.937317] [program2] : The child process has pid = 3930
[ 1005.937318] [program2] : This is the parent process, pid = 3929
[ 1005.937915] [program2] : child process
[ 1005.937916] [program2] : get SIGSTOP signal
[ 1005.937917] [program2] : child process Stop, unblockable (POSIX).
[ 1005.937918] [program2] : The return signal is 19
[ 1007.187867] [program2] : module_exit
```

```
[ 1156.649503] [program2] : module_init {Shaoqiang Sun} {120090638}
[ 1156.649505] [program2] : module_init create kthread start
[ 1156.649752] [program2] : module_init kthread start
[ 1156.649846] [program2] : The child process has pid = 4081
[ 1156.649847] [program2] : This is the parent process, pid = 4080
[ 1156.650521] [program2] : child process
[ 1156.650523] [program2] : child process Normal termination.
[ 1156.650524] [program2] : The return signal is 100
[ 1157.921731] [program2] : module_exit
```

What did you learn from the tasks?

In user mode:

I learn how to fork a process, how to print out the pid of the child process and the parent process, how to let child process execute test program, how to let parent process receive the SIGCHLD signal when child process finishes execution, how to check the status of the child process, and how to print out the termination information of child process according to the signal received.

I learn how to use fork(), execve(), waitpid(), WIFEXITED(), WIFSIGNALED(), WIFSTOPPED(), WEXITSTATUS(), WTERMSIG(), WSTOPSIG().

In kernel mode:

I learn how to create a kennel thread and how to fork a process, how to print out the pid of the child process and the parent process, how to let child process execute test program, how to let parent process wait until the child process terminates, and catches the signal raised in the child process, how to print out termination information of child process according to the signal raised in the child process.

I learn how to use kthread_create(), kernel_clone(), getname_kernel(), do_execve(), do_wait().