

A1 Report

120090521

庞嘉扬

How I design my program

In task1:

1. I fork a child process to execute test programs
2. Use `waitpid()` to let the parent process receives the `SIGCHLD` signal; also consider the `SIGSTOP` signal.
3. Use `switch` to judge the situation. Print out the termination information of child process.

In task2:

1. I create a kernel thread and run `my_fork()` function.
2. In `my_fork()` function I use `kernel_clone` to fork a process.
3. After forking a process into `my_exec`. In `my_exec` I actually called `do_exec` to run `test.o`. In this step I need to specify the absolute path of this test program.
4. Use `do_wait()` to let the parent process wait for the child process. Then the parent catch the signal returned by child. Print out related information.
5. The function `kernel_clone`, `do_execve`, `do_wait`, and `getname_kernel` is not visible to use because they are static in the source code. Here I modify the source code and use `export_symbol` to make them visible.

In bonus:

1. I use the function in `<dirent.h>` to get the information in `/proc` about `pid`
2. Then I get the information about `pid` in `/proc/pid/status` and `/proc/pid/task`, store them in struct
3. Build the tree according to the above information and print it

How I set my development environment.

1. Download the Linux source code; I use version 5.10.146 here.
2. Then decompression it in my virtual machine
3. Copy my previous config to the decompression path. Save the configuration.
4. Run make related command to build kernel Image and modules
5. Then install the kernel modules and install the kernel, reboot it. Finally, the new kernel is loaded.
6. To make some function in the linux source code visible, I modify them and recompile the kernel.

Secreenshots of my program output

Program1

```
root@csc3150:/home/vagrant/csc3150/Assignment_1_120090521/source/program1# ./program1 abort
Process start to fork
I'm the parent process, my pid = 4986
I'm the child process, my pid = 4987
Child process start to execute test program:
-----CHILD PROCESS START-----
This is the SIGABRT program

Parent process receives the SIGCHLD signal
child process get SIGABRT signal
child process is abort by abort signal
root@csc3150:/home/vagrant/csc3150/Assignment_1_120090521/source/program1# ./program1 alarm
Process start to fork
I'm the parent process, my pid = 5064
I'm the child process, my pid = 5065
Child process start to execute test program:
-----CHILD PROCESS START-----
This is the SIGALRM program

Parent process receives the SIGCHLD signal
child process get SIGALRM signal
child process is abort by alarm signal
root@csc3150:/home/vagrant/csc3150/Assignment_1_120090521/source/program1# ./program1 bus
Process start to fork
I'm the parent process, my pid = 5163
I'm the child process, my pid = 5164
Child process start to execute test program:
-----CHILD PROCESS START-----
This is the SIGBUS program

Parent process receives the SIGCHLD signal
child process get SIGBUS signal
child process is abort by bus error signal
root@csc3150:/home/vagrant/csc3150/Assignment_1_120090521/source/program1# ./program1 floating
Process start to fork
I'm the child process, my pid = 5209
Child process start to execute test program:
I'm the parent process, my pid = 5208
-----CHILD PROCESS START-----
This is the SIGFPE program

Parent process receives the SIGCHLD signal
child process get SIGFPE signal
child process is abort by floating error signal
root@csc3150:/home/vagrant/csc3150/Assignment_1_120090521/source/program1# ./program1 hangup
Process start to fork
I'm the parent process, my pid = 5262
I'm the child process, my pid = 5263
Child process start to execute test program:
-----CHILD PROCESS START-----
This is the SIGHUP program

Parent process receives the SIGCHLD signal
child process get SIGHUP signal
child process is abort by hung up signal
```

```

root@csc3150:/home/vagrant/csc3150/Assignment_1_120090521/source/program1# ./program1 illegal_instr
Process start to fork
I'm the parent process, my pid = 5372
I'm the child process, my pid = 5373
Child process start to execute test program:
-----CHILD PROCESS START-----
This is the SIGILL program

Parent process receives the SIGCHLD signal
child process get SIGILL signal
child process is abort by illegal signal
root@csc3150:/home/vagrant/csc3150/Assignment_1_120090521/source/program1# ./program1 kill
Process start to fork
I'm the parent process, my pid = 5417
I'm the child process, my pid = 5418
Child process start to execute test program:
-----CHILD PROCESS START-----
This is the SIGKILL program

Parent process receives the SIGCHLD signal
child process get SIGKILL signal
child process is killed
root@csc3150:/home/vagrant/csc3150/Assignment_1_120090521/source/program1# ./program1 normal
Process start to fork
I'm the parent process, my pid = 5461
I'm the child process, my pid = 5462
Child process start to execute test program:
-----CHILD PROCESS START-----
This is the normal program

-----CHILD PROCESS END-----
Parent process receives the SIGCHLD signal
Normal termination with EXIT STATUS = 0
root@csc3150:/home/vagrant/csc3150/Assignment_1_120090521/source/program1# ./program1 pipe
Process start to fork
I'm the parent process, my pid = 5547
I'm the child process, my pid = 5548
Child process start to execute test program:
-----CHILD PROCESS START-----
This is the SIGPIPE program

Parent process receives the SIGCHLD signal
child process get SIGPIPE signal
child process piped
root@csc3150:/home/vagrant/csc3150/Assignment_1_120090521/source/program1# ./program1 quit
Process start to fork
I'm the parent process, my pid = 5570
I'm the child process, my pid = 5571
Child process start to execute test program:
-----CHILD PROCESS START-----
This is the SIGQUIT program

Parent process receives the SIGCHLD signal
child process get SIGQUIT signal
child process quited

```

```

root@csc3150:/home/vagrant/csc3150/Assignment_1_120090521/source/program1# ./program1 segment_fault
Process start to fork
I'm the parent process, my pid = 5636
I'm the child process, my pid = 5637
Child process start to execute test program:
-----CHILD PROCESS START-----
This is the SIGSEGV program

Parent process receives the SIGCHLD signal
child process get SIGSEGV signal
child process is abort by segment fault signal
root@csc3150:/home/vagrant/csc3150/Assignment_1_120090521/source/program1# ./program1 stop
Process start to fork
I'm the parent process, my pid = 5702
I'm the child process, my pid = 5703
Child process start to execute test program:
-----CHILD PROCESS START-----
This is the SIGSTOP program

Parent process receives the SIGCHLD signal
child process get SIGSTOP signal
child execution stopped
root@csc3150:/home/vagrant/csc3150/Assignment_1_120090521/source/program1# ./program1 terminate
Process start to fork
I'm the parent process, my pid = 5767
I'm the child process, my pid = 5768
Child process start to execute test program:
-----CHILD PROCESS START-----
This is the SIGTERM program

Parent process receives the SIGCHLD signal
child process get SIGTERM signal
child process terminated
root@csc3150:/home/vagrant/csc3150/Assignment_1_120090521/source/program1# ./program1 trap
Process start to fork
I'm the parent process, my pid = 5790
I'm the child process, my pid = 5791
Child process start to execute test program:
-----CHILD PROCESS START-----
This is the SIGTRAP program

Parent process receives the SIGCHLD signal
child process get SIGTRAP signal
child process trapped
root@csc3150:/home/vagrant/csc3150/Assignment_1_120090521/source/program1# █

```

(A little different from my program1 code, this version is earlier than the inform by TA, but the TA said that there is no need to screenshot again. Thus, I just provide this old version here.)

Program2

```
[ 727.604299] [program2] : Module_init Jiayang Pang 120090521
[ 727.607487] [program2] : Module_init create kthread start
[ 727.609832] [program2] : Module_init kthread starts
[ 727.612272] [program2] : The child process has pid = 4236
[ 727.615738] [program2] : This is the parent process, pid = 4234
[ 727.616362] [program2] : child process
[ 727.953851] [program2] : get SIGBUS signal
[ 727.962024] [program2] : child process has bus error
[ 727.968761] [program2] : The return signal is 7
[ 730.952734] [program2] : Module_exit
```

bus

```
[ 890.929157] [program2] : Module_init Jiayang Pang 120090521
[ 890.935313] [program2] : Module_init create kthread start
[ 890.941073] [program2] : Module_init kthread starts
[ 890.946452] [program2] : The child process has pid = 7129
[ 890.949937] [program2] : This is the parent process, pid = 7128
[ 890.975213] [program2] : child process
[ 890.976394] [program2] : get SIGALARM signal
[ 890.980920] [program2] : child process has alarm error
[ 890.983487] [program2] : The return signal is 14
[ 893.050286] [program2] : Module_exit
```

alarm

```
[ 966.500310] [program2] : Module_init Jiayang Pang 120090521
[ 966.502667] [program2] : Module_init create kthread start
[ 966.505113] [program2] : Module_init kthread starts
[ 966.508763] [program2] : The child process has pid = 7967
[ 966.512783] [program2] : This is the parent process, pid = 7966
[ 966.513716] [program2] : child process
[ 966.808980] [program2] : get SIGFPE signal
[ 966.819286] [program2] : child process has float error
[ 966.824638] [program2] : The return signal is 8
[ 972.756060] [program2] : Module_exit
```

floating

abort

```
[ 928.083835] [program2] : Module_init Jiayang Pang 120090521
[ 928.087625] [program2] : Module_init create kthread start
[ 928.092255] [program2] : Module_init kthread starts
[ 928.096631] [program2] : The child process has pid = 7539
[ 928.100375] [program2] : This is the parent process, pid = 7538
[ 928.107549] [program2] : child process
[ 928.398481] [program2] : get SIGABRT signal
[ 928.410121] [program2] : child process has abort error
[ 928.416820] [program2] : The return signal is 6
[ 930.818215] [program2] : Module_exit
```

hangup

```
[ 1041.708370] [program2] : Module_init Jiayang Pang 120090521
[ 1041.712809] [program2] : Module_init create kthread start
[ 1041.717402] [program2] : Module_init kthread starts
[ 1041.723311] [program2] : The child process has pid = 8532
[ 1041.723687] [program2] : child process
[ 1041.728863] [program2] : This is the parent process, pid = 8531
[ 1041.738236] [program2] : get SIGHUP signal
[ 1041.742663] [program2] : child process is hung up
[ 1041.744978] [program2] : The return signal is 1
[ 1044.018601] [program2] : Module_exit
```

illegal

```
[ 1098.947510] [program2] : Module_init Jiayang Pang 120090521
[ 1098.952706] [program2] : Module_init create kthread start
[ 1098.956959] [program2] : Module_init kthread starts
[ 1098.960469] [program2] : The child process has pid = 9318
[ 1098.965234] [program2] : This is the parent process, pid = 9317
[ 1098.965485] [program2] : child process
[ 1098.968102] [program2] : get SIGINT signal
[ 1098.971093] [program2] : child process interrupt
[ 1098.975856] [program2] : The return signal is 2
[ 1101.154563] [program2] : Module_exit
```

interrupt

```
[ 1211.641055] [program2] : Module_init Jiayang Pang 120090521
[ 1211.643447] [program2] : Module_init create kthread start
[ 1211.646216] [program2] : Module_init kthread starts
[ 1211.653679] [program2] : The child process has pid = 9895
[ 1211.656208] [program2] : child process
[ 1211.657110] [program2] : This is the parent process, pid = 9894
[ 1211.663838] [program2] : get SIGKILL signal
[ 1211.666536] [program2] : child process killed
[ 1211.668509] [program2] : The return signal is 9
[ 1214.304270] [program2] : Module_exit
```

kill

```
[ 1243.133445] [program2] : Module_init Jiayang Pang 120090521  
[ 1243.136338] [program2] : Module_init create kthread start  
[ 1243.140763] [program2] : Module_init kthread starts  
[ 1243.143343] [program2] : The child process has pid = 10276  
[ 1243.147013] [program2] : This is the parent process, pid = 10275  
[ 1243.147985] [program2] : child process  
[ 1243.150352] [program2] : get SIGPIPE signal  
[ 1243.155843] [program2] : child process has pipe error  
[ 1243.157626] [program2] : The return signal is 13  
[ 1245.770866] [program2] : Module_exit
```

pipe

```
[ 1300.716609] [program2] : Module_init Jiayang Pang 120090521  
[ 1300.722740] [program2] : Module_init create kthread start  
[ 1300.726732] [program2] : Module_init kthread starts  
[ 1300.730616] [program2] : The child process has pid = 10781  
[ 1300.730708] [program2] : child process  
[ 1300.735222] [program2] : This is the parent process, pid = 10779  
[ 1300.997334] [program2] : get SIGQUIT signal  
[ 1301.001521] [program2] : child process quit  
[ 1301.005105] [program2] : The return signal is 3  
[ 1303.138514] [program2] : Module_exit
```

quit

```
[ 1350.070725] [program2] : Module_init Jiayang Pang 120090521  
[ 1350.073704] [program2] : Module_init create kthread start  
[ 1350.076120] [program2] : Module_init kthread starts  
[ 1350.080263] [program2] : The child process has pid = 11222  
[ 1350.082647] [program2] : child process  
[ 1350.082860] [program2] : This is the parent process, pid = 11221  
[ 1350.375449] [program2] : get SIGSEGV signal  
[ 1350.377660] [program2] : child process has segmentation fault error  
[ 1350.381968] [program2] : The return signal is 11  
[ 1352.067011] [program2] : Module_exit
```

segmentation


```
[ 1374.753909] [program2] : Module_init Jiayang Pang 120090521
[ 1374.760327] [program2] : Module_init create kthread start
[ 1374.767853] [program2] : Module_init kthread starts
[ 1374.773658] [program2] : The child process has pid = 11620
[ 1374.779231] [program2] : This is the parent process, pid = 11619
[ 1374.779844] [program2] : child process
[ 1374.783297] [program2] : get SIGTERM signal
[ 1374.791923] [program2] : child process terminated
[ 1374.794843] [program2] : The return signal is 15
[ 1377.044184] [program2] : Module_exit
```

terminated

```
[ 1404.235539] [program2] : Module_init Jiayang Pang 120090521
[ 1404.238227] [program2] : Module_init create kthread start
[ 1404.240335] [program2] : Module_init kthread starts
[ 1404.243754] [program2] : The child process has pid = 12055
[ 1404.246527] [program2] : This is the parent process, pid = 12054
[ 1404.246887] [program2] : child process
[ 1404.490497] [program2] : get SIGTRAP signal
[ 1404.498321] [program2] : child process has trap error
[ 1404.504959] [program2] : The return signal is 5
[ 1405.904483] [program2] : Module_exit
```

trap

```
[ 3161.716549] [program2] : Module_init Jiayang Pang 120090521
[ 3161.722668] [program2] : Module_init create kthread start
[ 3161.728689] [program2] : Module_init kthread starts
[ 3161.732632] [program2] : The child process has pid = 15234
[ 3161.734762] [program2] : child process
[ 3161.737977] [program2] : This is the parent process, pid = 15232
[ 3161.754822] [program2] : get SIGSTOP signal
[ 3161.757434] [program2] : child process has stopped
[ 3161.762000] [program2] : The return signal is 19
[ 3163.739500] [program2] : Module_exit
```

stop

Bonus

(there are some shortcomings in my code, like there are no threads shown in the graph)

```

vagrant@csc3150:~/csc3150/Assignment_1_120090521/source/bonus$ ./pstree
systemd+-systemd-(sd-pam)
|-irqbalance
|-agetty
|-agetty
|-mdadm
|-polkitd
|-unattended-upgr
|-iscsid
|-iscsid
|-sshd-sshd-sshd-bash-bash-sh-node+-node
|                                     |-node
|                                     |-node-bash-pstree
|-dbus-daemon
|-acpid
|-systemd-logind
|-accounts-daemon
|-lxcfs
|-atd
|-rsyslogd
|-cron
|-dhclient
|-systemd-udev
|-lvmetad
|-systemd-journal

```

```

vagrant@csc3150:~/csc3150/Assignment_1_120090521/source/bonus$ ./pstree -p
systemd(1)+-systemd(1411)-(sd-pam)(1412)
|-irqbalance(1075)
|-agetty(1052)
|-agetty(1046)
|-mdadm(1044)
|-polkitd(1024)
|-unattended-upgr(1014)
|-iscsid(1007)
|-iscsid(1006)
|-sshd(1004)-sshd(1409)-sshd(1448)-bash(1449)-bash(1453)-sh(1498)-node(1508)+-node(1637)
|                                     |-node(1626)
|                                     |-node(1593)-bash(4905)-pstree(5032)
|-dbus-daemon(983)
|-acpid(979)
|-systemd-logind(975)
|-accounts-daemon(973)
|-lxcfs(972)
|-atd(966)
|-rsyslogd(961)
|-cron(959)
|-dhclient(807)
|-systemd-udev(399)
|-lvmetad(391)
|-systemd-journal(362)

```

```

vagrant@csc3150:~/csc3150/Assignment_1_120090521/source/bonus$ ./pstree -g
systemd(1)─systemd(1411)─(sd-pam)(1411)
    │
    ├─irqbalance(1075)
    ├─agetty(1052)
    ├─agetty(1046)
    ├─mdadm(1044)
    ├─polkitd(1024)
    ├─unattended-upgr(1014)
    ├─iscsid(1007)
    ├─iscsid(1006)
    ├─sshd(1004)─sshd(1409)─sshd(1409)─bash(1449)─bash(1449)─sh(1449)─node(1449)─node(1449)
    │                                                    │
    │                                                    └─node(1449)─bash(4905)─pstree(5057)
    │
    ├─dbus-daemon(983)
    ├─acpid(979)
    ├─systemd-logind(975)
    ├─accounts-daemon(973)
    ├─lxcfs(972)
    ├─atd(966)
    ├─rsyslogd(961)
    ├─cron(959)
    ├─dhclient(807)
    ├─systemd-udev(399)
    ├─lvmetad(391)
    └─systemd-journal(362)

```

```

vagrant@csc3150:~/csc3150/Assignment_1_120090521/source/bonus$ ./pstree -v
pstree (PSmisc) UNKNOWN
Copyright (C) 1993-2019 Werner Almesberger and Craig Small
PSmisc comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it under
the terms of the GNU General Public License.
For more information about these matters, see the files named COPYING.

```

Pstree -A and pstree -c are not shown because they are the same as the first screenshot pstree

What did I learn from this task

1. Learn how the program run in user mode and kernel mode. How the process fork and return signal to its parent.
2. How to set the Linux environment.
3. More adept at Linux command and Makefile.
4. Learn some basic functions in the source Linux code and try to read some source code to understand their relationships.
5. For the bonus part, learn how to get information about the processes in c.