Report for Project 1

Name: Ziteng Yang

1. **Project Files**

./

Prj1-517021910683.pdf [Description: this file]

case-study [Description: the example program]

README [Description: list of files]

problem-1

Makefile

ptree_syscall.c [Description: system call that pass an array of process information to user process in DFS order]

problem-2

ptree

jni

Android.mk

ptree.c [Description: the user program that invoke system call to get an array of process information and print to screen in DFS order]

problem-3

testptree

jni

Android.mk

testptree.c [Description: the user program that create a child process to invoke ptree program and print the relation ship of parent and child process]

problem-4

server.c [Description: the code of the sercer program]

client.c [Decription: the code of the client program]

testscript

testscript_problem_1&2 [Description: an example of the result of calling ptree in terminal]

testscript_problem_3 [Description: an example of the result of calling ptree in another test program]

testscript_problem_4 [Description: containing an example of three client communication with one server]

client1

client2

client3

server

README

2. Process Description

System call of "ptree_syscall.c", as required, uses DFS to explore the current process, and return the information of them in DFS order, it also use "printk(/****/)" function to output the tree in dmesg screen. I used an int varieble “depth” to denote how many tabs we need to print on the screen to denote a tree.

This system call will explore every current process, starting from the root process with pid=0, and using the data structure to visit its every child and child’s child and so on, totally in DFS order, and pass the information to the pointer passed from user program.

It also prints the information on “dmesg” screen.

Program of file "ptree.c" tests the new module by call system call (numbered 356), and pass a pointer to the module, getting the result and output them by "printf(/****/)" function, also in DFS order.

The executable file it generates is named "ptree".

One example output is included in the file "testscripy/testscript_problem_1&2".

Program of "testptree.c" create a child process, which calls "ptree" in problem-2, then wait for child process to execute exit(). Then it output both its PID as Parent PID and its child's PID as Child PID and my studentID as required. The ptree called in child process also shows the relationship between parent process and child process as required.

The executable file it generates is named "testptree".

One example output is included in the file "testscripy/testscript_problem_3".

Program of file “server.c” connects clients that want to have connection (at most 5, otherwise wait), and create a thread for each clien.t If 2 clients is being served, then for the third client, the server receive the information but only sent “Please wait!” without performing an encoding job.

Mutex was used to protect the incrementing and decreasing the value of client counter.

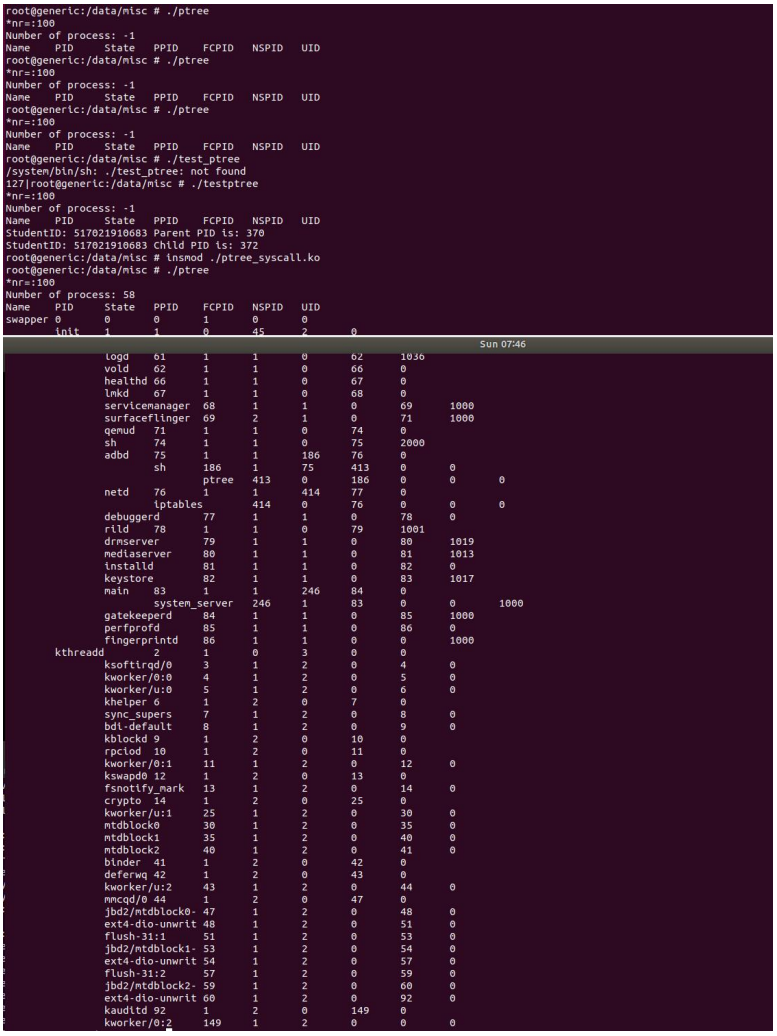
Note that the sequence of serving is not dependent on who connected first, but on who send the first sentence first.

I also created a thread for operating servers to close safely, rather than needing to use “ctrl+z”. It’s not so perfect but can **reluctantly** work in some way.

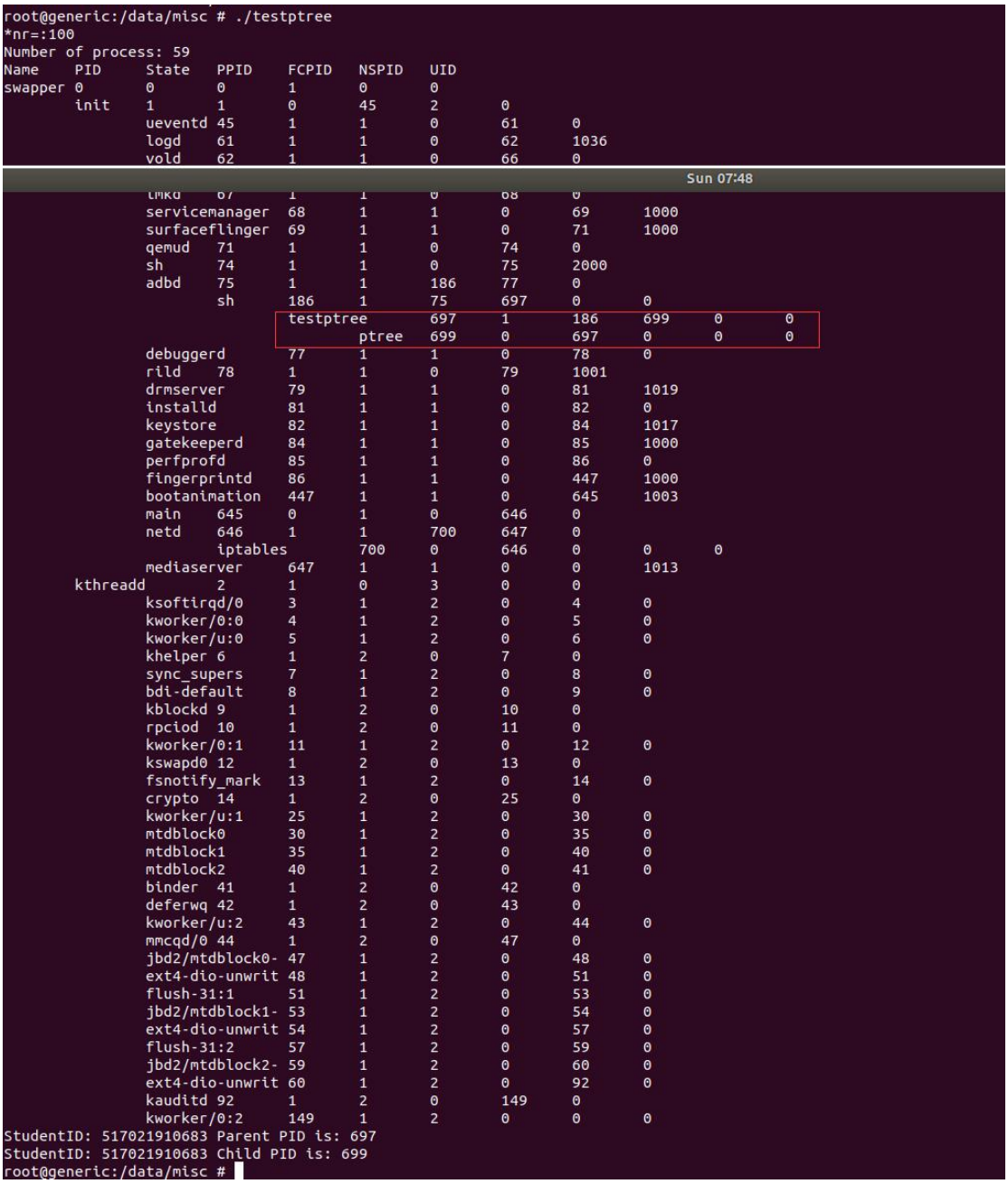
Program of file “client.c” connects with server, gets inputs from user and then send them to server. If the input is “:q”, it breaks the loop (“:q” will also be send to server as a signal).

3. Result

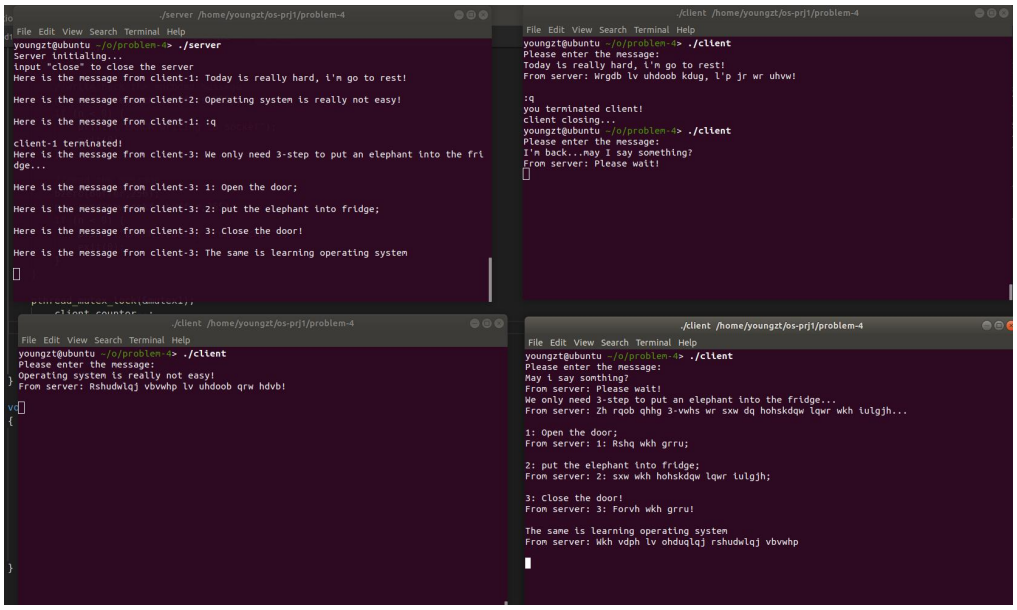
3.1. Problem 1 & Problem 2


root@generic:/data/misc # ./ptree
*nr=:100
Number of process: -1
Name PID State PPID FCPID NSPID UID
root@generic:/data/misc # ./ptree
*nr=:100
Number of process: -1
Name PID State PPID FCPID NSPID UID
root@generic:/data/misc # ./ptree
*nr=:100
Number of process: -1
Name PID State PPID FCPID NSPID UID
root@generic:/data/misc # ./test_ptree
/system/bin/sh: ./test_ptree: not found
127|root@generic:/data/misc # ./testptree
*nr=:100
Number of process: -1
Name PID State PPID FCPID NSPID UID
StudentID: 517021910683 Parent PID is: 370
StudentID: 517021910683 Child PID is: 372
root@generic:/data/misc # insmod ./ptree_syscall.ko
root@generic:/data/misc # ./ptree
*nr=:100
Number of process: 58
Name PID State PPID FCPID NSPID UID
swapper 0 0 0 1 0 0
init 1 1 0 45 2 0
Sun 07:46
logd 61 1 1 0 62 1036
vold 62 1 1 0 66 0
healthd 66 1 1 0 67 0
lmkd 67 1 1 0 68 0
servicemanager 68 1 1 0 69 1000
surfaceflinger 69 2 1 0 71 1000
qemud 71 1 1 0 74 0
sh 74 1 1 0 75 2000
adbd 75 1 1 186 76 0
sh 186 1 75 413 0 0
ptree 413 0 186 0 0 0
netd 76 1 1 414 77 0
iptables 414 0 76 0 0 0
debuggerd 77 1 1 0 78 0
rild 78 1 1 0 79 1001
drmserver 79 1 1 0 80 1019
mediaserver 80 1 1 0 81 1013
installd 81 1 1 0 82 0
keystore 82 1 1 0 83 1017
main 83 1 1 246 84 0
system_server 246 1 83 0 0 1000
gatekeeperd 84 1 1 0 85 1000
perfprofd 85 1 1 0 86 0
fingerprintd 86 1 1 0 0 1000
kthreadd 2 1 0 3 0 0
ksoftirqd/0 3 1 2 0 4 0
kworker/0:0 4 1 2 0 5 0
kworker/u:0 5 1 2 0 6 0
khelper 6 1 2 0 7 0
sync_supers 7 1 2 0 8 0
bdi-default 8 1 2 0 9 0
kblockd 9 1 2 0 10 0
rpciod 10 1 2 0 11 0
kworker/0:1 11 1 2 0 12 0
kswapd0 12 1 2 0 13 0
fsnotify_mark 13 1 2 0 14 0
crypto 14 1 2 0 25 0
kworker/u:1 25 1 2 0 30 0
mtdblock0 30 1 2 0 35 0
mtdblock1 35 1 2 0 40 0
mtdblock2 40 1 2 0 41 0
binder 41 1 2 0 42 0
deferwq 42 1 2 0 43 0
kworker/u:2 43 1 2 0 44 0
mmcqd/0 44 1 2 0 47 0
jbd2/mtdblock0- 47 1 2 0 48 0
ext4-dio-unwrit 48 1 2 0 51 0
flush-31:1 51 1 2 0 53 0
jbd2/mtdblock1- 53 1 2 0 54 0
ext4-dio-unwrit 54 1 2 0 57 0
flush-31:2 57 1 2 0 59 0
jbd2/mtdblock2- 59 1 2 0 60 0
ext4-dio-unwrit 60 1 2 0 92 0
kaudltd 92 1 2 0 149 0
kworker/0:2 149 1 2 0 0 0

3.2. Problem 3


root@generic:/data/misc # ./testptree
*nr=:100
Number of process: 59
Name PID State PPID FCPID NSPID UID
swapper 0 0 0 1 0 0
init 1 1 0 45 2 0
ueventd 45 1 1 0 61 0
logd 61 1 1 0 62 1036
vold 62 1 1 0 66 0
Sun 07:48
tmkd 67 1 1 0 68 0
servicemanager 68 1 1 0 69 1000
surfaceflinger 69 1 1 0 71 1000
qemud 71 1 1 0 74 0
sh 74 1 1 0 75 2000
adbd 75 1 1 186 77 0
sh 186 1 75 697 0 0
testptree 697 1 186 699 0 0
ptree 699 0 697 0 0 0
debuggerd 77 1 1 0 78 0
rild 78 1 1 0 79 1001
drmserver 79 1 1 0 81 1019
installd 81 1 1 0 82 0
keystore 82 1 1 0 84 1017
gatekeeperd 84 1 1 0 85 1000
perfprofd 85 1 1 0 86 0
fingerprintd 86 1 1 0 447 1000
bootanimation 447 1 1 0 645 1003
main 645 0 1 0 646 0
netd 646 1 1 700 647 0
iptables 700 0 646 0 0 0
mediaserver 647 1 1 0 0 1013
kthreadd 2 1 0 3 0 0
ksoftirqd/0 3 1 2 0 4 0
kworker/0:0 4 1 2 0 5 0
kworker/u:0 5 1 2 0 6 0
khelper 6 1 2 0 7 0
sync_supers 7 1 2 0 8 0
bdi-default 8 1 2 0 9 0
kblockd 9 1 2 0 10 0
rpciod 10 1 2 0 11 0
kworker/0:1 11 1 2 0 12 0
kswapd0 12 1 2 0 13 0
fsnotify_mark 13 1 2 0 14 0
crypto 14 1 2 0 25 0
kworker/u:1 25 1 2 0 30 0
mtdblock0 30 1 2 0 35 0
mtdblock1 35 1 2 0 40 0
mtdblock2 40 1 2 0 41 0
binder 41 1 2 0 42 0
deferwq 42 1 2 0 43 0
kworker/u:2 43 1 2 0 44 0
mmcqd/0 44 1 2 0 47 0
jbd2/mtdblock0- 47 1 2 0 48 0
ext4-dio-unwrit 48 1 2 0 51 0
flush-31:1 51 1 2 0 53 0
jbd2/mtdblock1- 53 1 2 0 54 0
ext4-dio-unwrit 54 1 2 0 57 0
flush-31:2 57 1 2 0 59 0
jbd2/mtdblock2- 59 1 2 0 60 0
ext4-dio-unwrit 60 1 2 0 92 0
kauditd 92 1 2 0 149 0
kworker/0:2 149 1 2 0 0 0
StudentID: 517021910683 Parent PID is: 697
StudentID: 517021910683 Child PID is: 699
root@generic:/data/misc #

3.3. Problem 4


./server /home/youngzt/os-prj1/problem-4
File Edit View Search Terminal Help
youngzt@ubuntu ~/o/problem-4> ./server
Server initialing...
input "close" to close the server
Here is the message from client-1: Today is really hard, i'm go to rest!
Here is the message from client-2: Operating system is really not easy!
Here is the message from client-1: :q
client-1 terminated!
Here is the message from client-3: We only need 3-step to put an elephant into the fridge...
Here is the message from client-3: 1: Open the door;
Here is the message from client-3: 2: put the elephant into fridge;
Here is the message from client-3: 3: Close the door!
Here is the message from client-3: The same is learning operating system
./client /home/youngzt/os-prj1/problem-4
File Edit View Search Terminal Help
youngzt@ubuntu ~/o/problem-4> ./client
Please enter the message:
Today is really hard, i'm go to rest!
From server: Wrgdb lv uhdoob kdug, i'p jr wr uhvw!
:q
you terminated client!
client closing...
youngzt@ubuntu ~/o/problem-4> ./client
Please enter the message:
I'm back...may I say something?
From server: Please wait!
./client /home/youngzt/os-prj1/problem-4
File Edit View Search Terminal Help
youngzt@ubuntu ~/o/problem-4> ./client
Please enter the message:
Operating system is really not easy!
From server: Rshudwlqj vbvwhp lv uhdoob qrw hdvb!
./client /home/youngzt/os-prj1/problem-4
File Edit View Search Terminal Help
youngzt@ubuntu ~/o/problem-4> ./client
Please enter the message:
May i say somthing?
From server: Please wait!
We only need 3-step to put an elephant into the fridge...
From server: Zh rqob qhhg 3-vwhs wr sxw dq hohskdqw lqwr wkh iulgjh...
1: Open the door;
From server: 1: Rshq wkh grru;
2: put the elephant into fridge;
From server: 2: sxw wkh hohskdqw lqwr iulgjh;
3: Close the door!
From server: 3: Forvh wkh grru!
The same is learning operating system
From server: Wkh vdph lv ohduqlqj rshudwlqj vbvwhp