CIS 655 Advanced Computer Architecture
Fall Final Report

Group Member
Name (SUID)
LIU, SHOU YUNG (514096323)
Zezhi Su(368202764)

1. Project Scope

In the final project, we modify the project scope to match the project requirement. Below are the target which we focus on.

1. Write a module to intercept all the page faults and send signal to the user process.
2. Find the mechanism to trigger original page fault function

2. Project member

SUID	Name	Email
368202764	Zezhi Su	zsu103@syr.edu
514096323	LIU, Shou Yung	sliu132@syr.edu

3. Project Description

1. Introduction

   Following the project requirements, we need to find the mechanism to intercept the page fault occur, and also trigger original page fault mechanism. The difficult point of the project is "How to invoke the original do page fault function without modify the original kernel module?" If we could modify the original kernel function, then it will be easier to achieve what we expect to. Modifying the do page fault function and declare the function to extern, then write another Linux Kernel Module to call the original page fault function after print the page fault address detail information. But if we want to get the page fault information also trigger original do page fault, we have to find another way to achieve this. Our solution is reading the function from /proc/kallsyms, which is the Linux Kernel global symbol table. Extract the non-export function address from the symbol table, so that we could execute the original page fault mechanism. Next section describe the purpose of the file. The third section display the execution screen shot and code explanation.

2. Project file Description

File List	Description
load.sh	1. Load the kernel function from the **boot/System.map-${LINUX_VERSION}**, which could correctly execute the whole project. 2. Linked compiled module to the kernel
unload.sh	Remove linked module from the kernel
makefile	1. Compile customPageFault.c, initPageFaultListener.c two file into intercept.o, also generate intercept.ko, intercept.mod.c, module.order, files which are needed for the kernel module. 2. Clean compiled module
initPageFaultListener.c	Include two function- init_page_fault_listener, exit_page_fault_listener 1. init_page_fault_listener: register customPageFault function. 2. exit_page_fault_listener: remove kernel module
customPageFault.c	1. registerPageFaultListener: ● Register page fault function. Receive parameters which comes from symbol table. ● Store default idtr ● Get the idtr default address ● Allocate new page for new idtr ● Copy the old idtr to new page

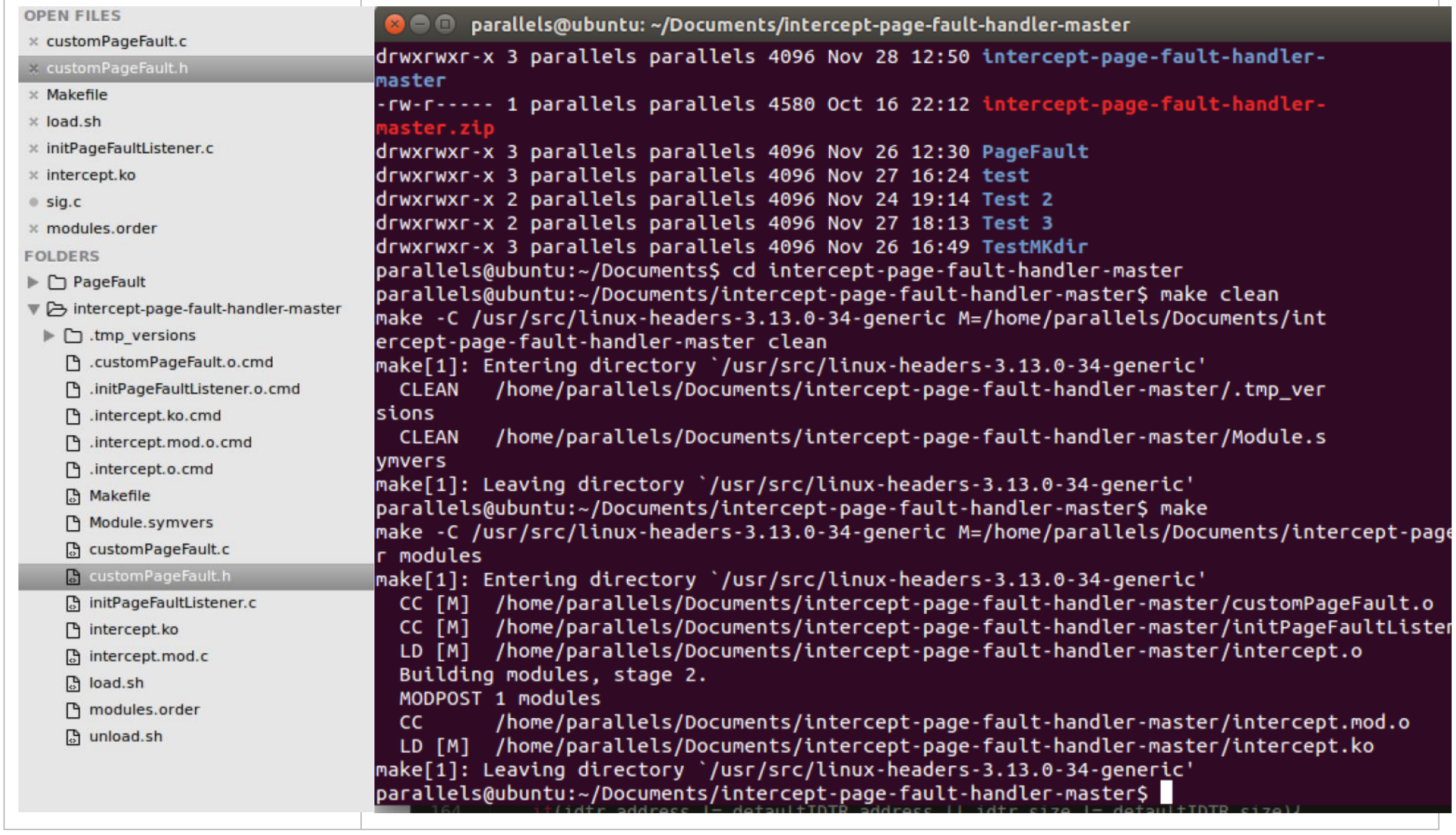
	● Loaded idt to all the cpu 2. unregisterPageFaultListener: ● restore default idtr and free the allocate page 3. write_pid ● Get user space process id from then send message to it
customPageFault.h	customPageFault.h Header file
UserSpaceCode.c	UserSpace process which could receive the kernel information.

3. Execution Step

i. Under Ubuntu 3.13.0-34-generic X86_64 platform open terminal

ii. Change directory to the project directory

iii. Type “make”

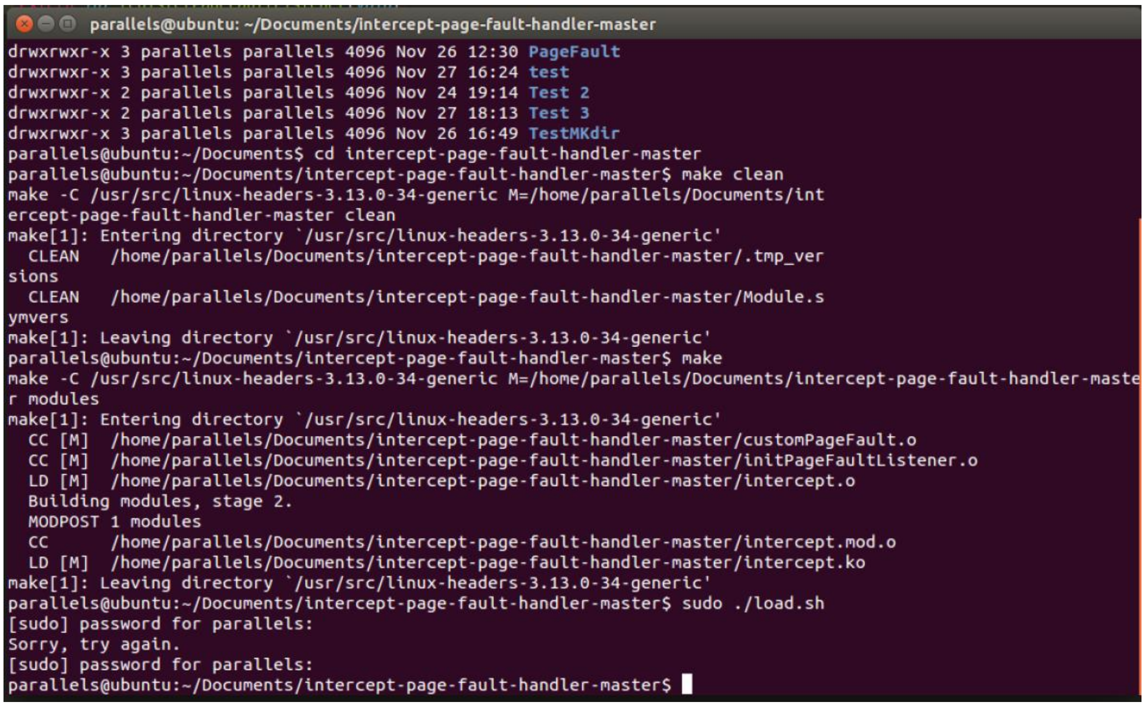
Make result	Step from 1 to 3
	

```
OPEN FILES
× customPageFault.c
× customPageFault.h
× Makefile
× load.sh
× initPageFaultListener.c
× intercept.ko
● sig.c
× modules.order
FOLDERS
▶ PageFault
▼ intercept-page-fault-handler-master
  ▶ .tmp_versions
    .customPageFault.o.cmd
    .initPageFaultListener.o.cmd
    .intercept.ko.cmd
    .intercept.mod.o.cmd
    .intercept.o.cmd
    Makefile
    Module.symvers
    customPageFault.c
    customPageFault.h
    initPageFaultListener.c
    intercept.ko
    intercept.mod.c
    load.sh
    modules.order
    unload.sh
```

```
parallels@ubuntu: ~/Documents/intercept-page-fault-handler-master
drwxrwxr-x 3 parallels parallels 4096 Nov 28 12:50 intercept-page-fault-handler-
master
-rw-r----- 1 parallels parallels 4580 Oct 16 22:12 intercept-page-fault-handler-
master.zip
drwxrwxr-x 3 parallels parallels 4096 Nov 26 12:30 PageFault
drwxrwxr-x 3 parallels parallels 4096 Nov 27 16:24 test
drwxrwxr-x 2 parallels parallels 4096 Nov 24 19:14 Test 2
drwxrwxr-x 2 parallels parallels 4096 Nov 27 18:13 Test 3
drwxrwxr-x 3 parallels parallels 4096 Nov 26 16:49 TestMKdir
parallels@ubuntu:~/Documents$ cd intercept-page-fault-handler-master
parallels@ubuntu:~/Documents/intercept-page-fault-handler-master$ make clean
make -C /usr/src/linux-headers-3.13.0-34-generic M=/home/parallels/Documents/int
ercept-page-fault-handler-master clean
make[1]: Entering directory `/usr/src/linux-headers-3.13.0-34-generic'
  CLEAN   /home/parallels/Documents/intercept-page-fault-handler-master/.tmp_ver
sions
  CLEAN   /home/parallels/Documents/intercept-page-fault-handler-master/Module.s
ymvers
make[1]: Leaving directory `/usr/src/linux-headers-3.13.0-34-generic'
parallels@ubuntu:~/Documents/intercept-page-fault-handler-master$ make
make -C /usr/src/linux-headers-3.13.0-34-generic M=/home/parallels/Documents/intercept-page
r modules
make[1]: Entering directory `/usr/src/linux-headers-3.13.0-34-generic'
  CC [M]  /home/parallels/Documents/intercept-page-fault-handler-master/customPageFault.o
  CC [M]  /home/parallels/Documents/intercept-page-fault-handler-master/initPageFaultListe
  LD [M]  /home/parallels/Documents/intercept-page-fault-handler-master/intercept.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /home/parallels/Documents/intercept-page-fault-handler-master/intercept.mod.o
  LD [M]  /home/parallels/Documents/intercept-page-fault-handler-master/intercept.ko
make[1]: Leaving directory `/usr/src/linux-headers-3.13.0-34-generic'
parallels@ubuntu:~/Documents/intercept-page-fault-handler-master$ 
```

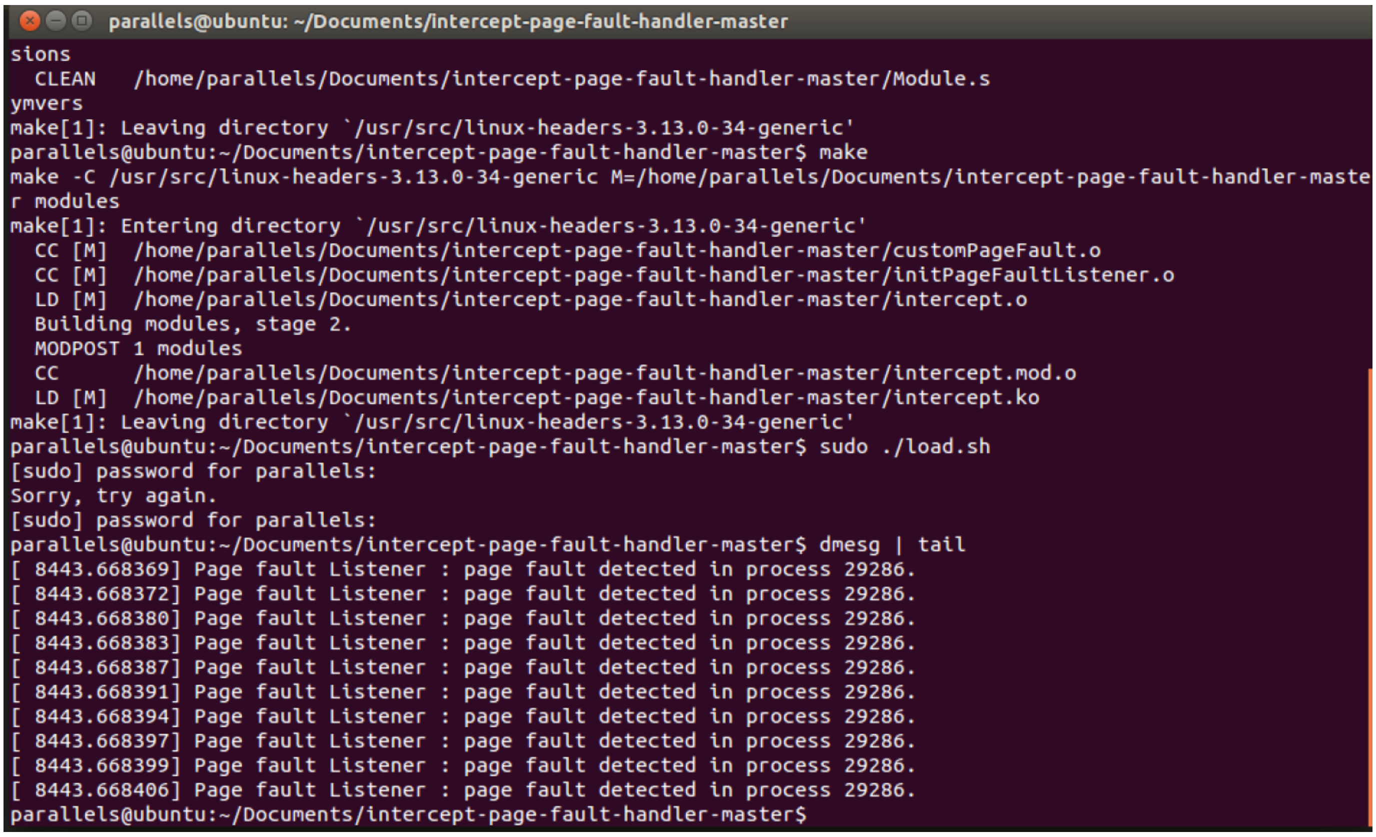
iv. Type sudo ./load.sh

v. Successful insert module from it



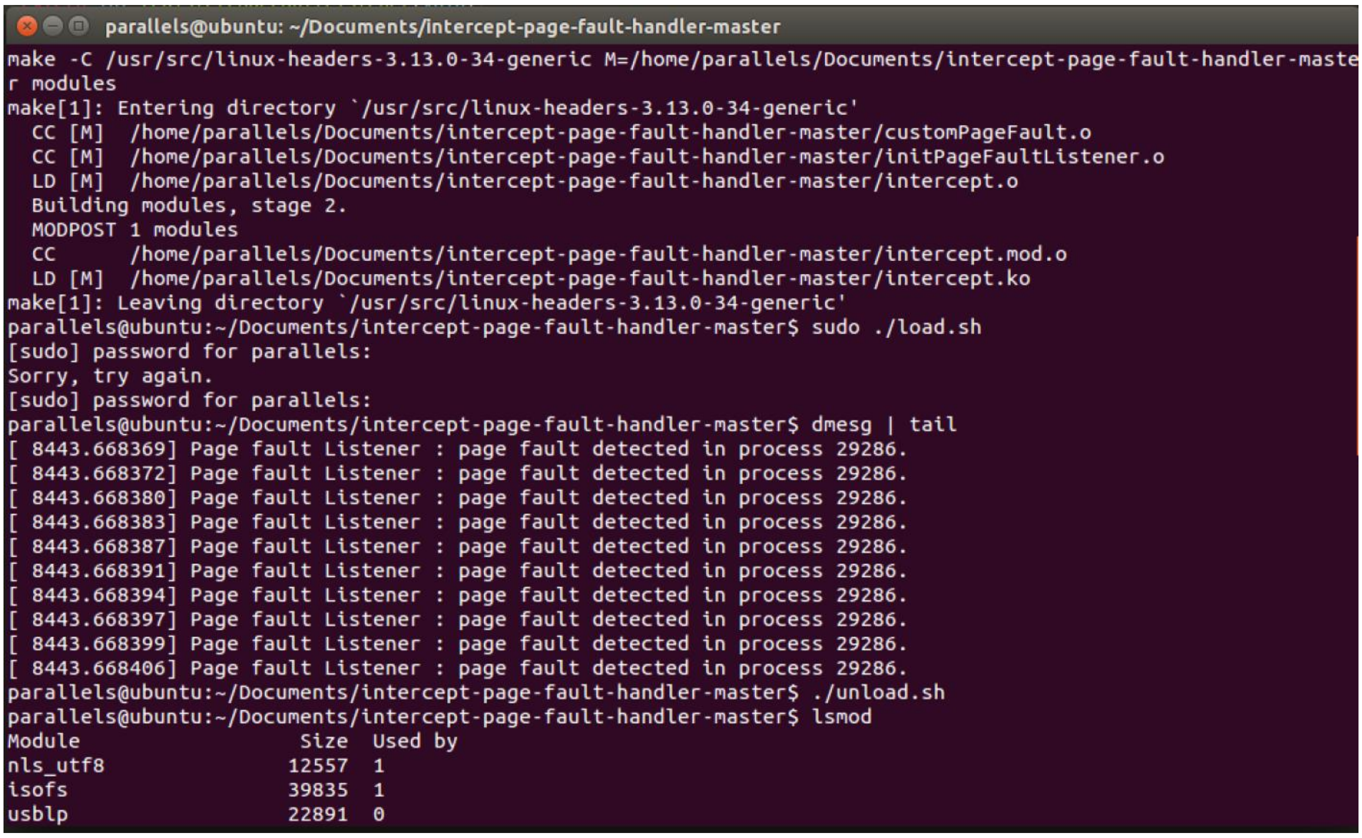
```
parallels@ubuntu: ~/Documents/intercept-page-fault-handler-master
drwxrwxr-x 3 parallels parallels 4096 Nov 26 12:30 PageFault
drwxrwxr-x 3 parallels parallels 4096 Nov 27 16:24 test
drwxrwxr-x 2 parallels parallels 4096 Nov 24 19:14 Test 2
drwxrwxr-x 2 parallels parallels 4096 Nov 27 18:13 Test 3
drwxrwxr-x 3 parallels parallels 4096 Nov 26 16:49 TestMKdir
parallels@ubuntu:~/Documents$ cd intercept-page-fault-handler-master
parallels@ubuntu:~/Documents/intercept-page-fault-handler-master$ make clean
make -C /usr/src/linux-headers-3.13.0-34-generic M=/home/parallels/Documents/int
ercept-page-fault-handler-master clean
make[1]: Entering directory `/usr/src/linux-headers-3.13.0-34-generic'
  CLEAN   /home/parallels/Documents/intercept-page-fault-handler-master/.tmp_ver
sions
  CLEAN   /home/parallels/Documents/intercept-page-fault-handler-master/Module.s
ymvers
make[1]: Leaving directory `/usr/src/linux-headers-3.13.0-34-generic'
parallels@ubuntu:~/Documents/intercept-page-fault-handler-master$ make
make -C /usr/src/linux-headers-3.13.0-34-generic M=/home/parallels/Documents/intercept-page-fault-handler-maste
r modules
make[1]: Entering directory `/usr/src/linux-headers-3.13.0-34-generic'
  CC [M]  /home/parallels/Documents/intercept-page-fault-handler-master/customPageFault.o
  CC [M]  /home/parallels/Documents/intercept-page-fault-handler-master/initPageFaultListener.o
  LD [M]  /home/parallels/Documents/intercept-page-fault-handler-master/intercept.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /home/parallels/Documents/intercept-page-fault-handler-master/intercept.mod.o
  LD [M]  /home/parallels/Documents/intercept-page-fault-handler-master/intercept.ko
make[1]: Leaving directory `/usr/src/linux-headers-3.13.0-34-generic'
parallels@ubuntu:~/Documents/intercept-page-fault-handler-master$ sudo ./load.sh
[sudo] password for parallels:
Sorry, try again.
[sudo] password for parallels:
parallels@ubuntu:~/Documents/intercept-page-fault-handler-master$ 
```

vi. Type “dmesg | tail” to show the printk page fault information



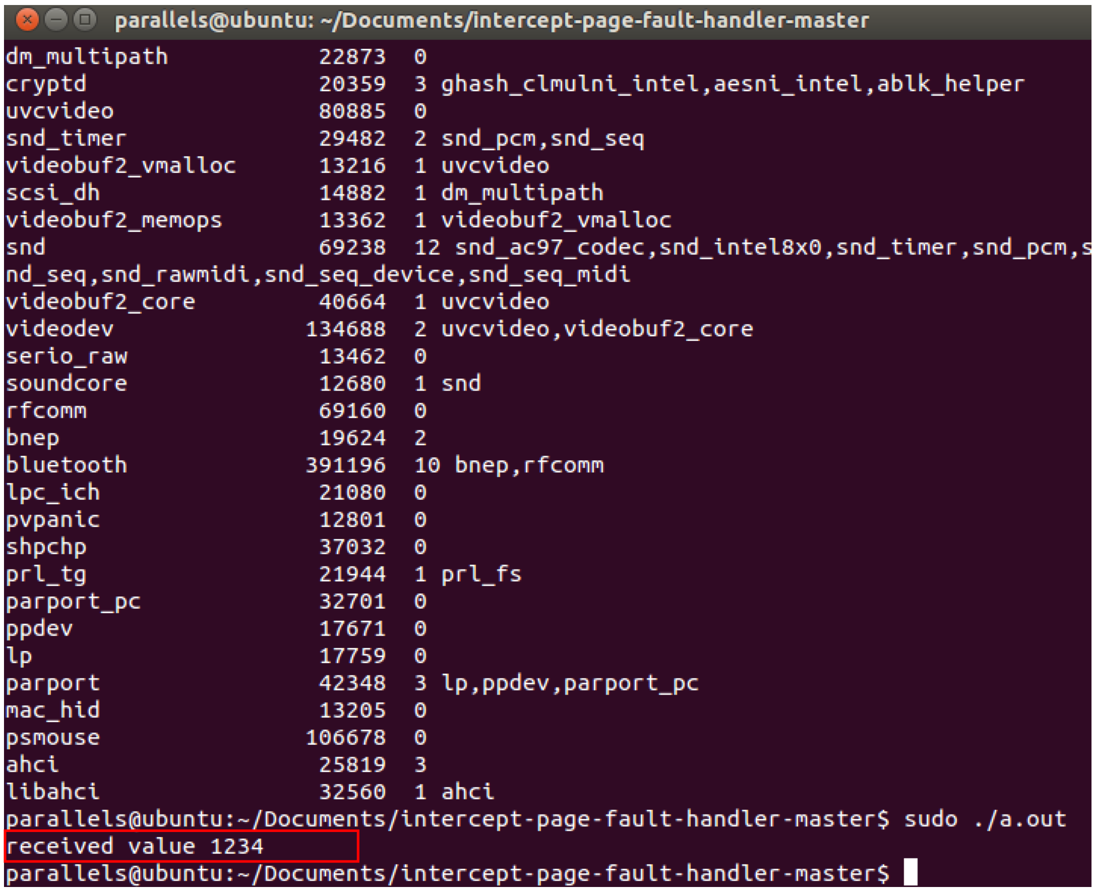
```
parallels@ubuntu: ~/Documents/intercept-page-fault-handler-master
sions
  CLEAN   /home/parallels/Documents/intercept-page-fault-handler-master/Module.s
ymvers
make[1]: Leaving directory `/usr/src/linux-headers-3.13.0-34-generic'
parallels@ubuntu:~/Documents/intercept-page-fault-handler-master$ make
make -C /usr/src/linux-headers-3.13.0-34-generic M=/home/parallels/Documents/intercept-page-fault-handler-maste
r modules
make[1]: Entering directory `/usr/src/linux-headers-3.13.0-34-generic'
  CC [M]  /home/parallels/Documents/intercept-page-fault-handler-master/customPageFault.o
  CC [M]  /home/parallels/Documents/intercept-page-fault-handler-master/initPageFaultListener.o
  LD [M]  /home/parallels/Documents/intercept-page-fault-handler-master/intercept.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /home/parallels/Documents/intercept-page-fault-handler-master/intercept.mod.o
  LD [M]  /home/parallels/Documents/intercept-page-fault-handler-master/intercept.ko
make[1]: Leaving directory `/usr/src/linux-headers-3.13.0-34-generic'
parallels@ubuntu:~/Documents/intercept-page-fault-handler-master$ sudo ./load.sh
[sudo] password for parallels:
Sorry, try again.
[sudo] password for parallels:
parallels@ubuntu:~/Documents/intercept-page-fault-handler-master$ dmesg | tail
[ 8443.668369] Page fault Listener : page fault detected in process 29286.
[ 8443.668372] Page fault Listener : page fault detected in process 29286.
[ 8443.668380] Page fault Listener : page fault detected in process 29286.
[ 8443.668383] Page fault Listener : page fault detected in process 29286.
[ 8443.668387] Page fault Listener : page fault detected in process 29286.
[ 8443.668391] Page fault Listener : page fault detected in process 29286.
[ 8443.668394] Page fault Listener : page fault detected in process 29286.
[ 8443.668397] Page fault Listener : page fault detected in process 29286.
[ 8443.668399] Page fault Listener : page fault detected in process 29286.
[ 8443.668406] Page fault Listener : page fault detected in process 29286.
parallels@ubuntu:~/Documents/intercept-page-fault-handler-master$
```

vii. Type ./unload.sh to remove module, and use “lsmod” check the module is indeed remove



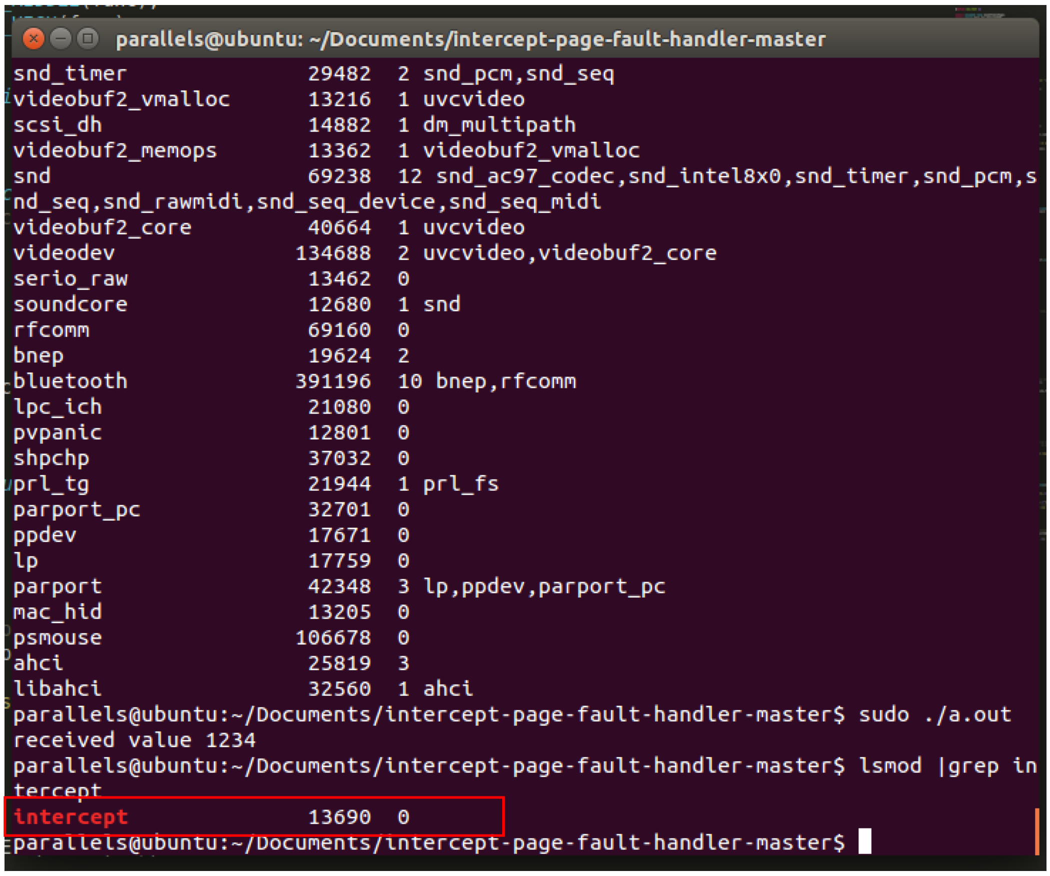
```
make -C /usr/src/linux-headers-3.13.0-34-generic M=/home/parallels/Documents/intercept-page-fault-handler-maste
r modules
make[1]: Entering directory `/usr/src/linux-headers-3.13.0-34-generic'
  CC [M]  /home/parallels/Documents/intercept-page-fault-handler-master/customPageFault.o
  CC [M]  /home/parallels/Documents/intercept-page-fault-handler-master/initPageFaultListener.o
  LD [M]  /home/parallels/Documents/intercept-page-fault-handler-master/intercept.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /home/parallels/Documents/intercept-page-fault-handler-master/intercept.mod.o
  LD [M]  /home/parallels/Documents/intercept-page-fault-handler-master/intercept.ko
make[1]: Leaving directory `/usr/src/linux-headers-3.13.0-34-generic'
parallels@ubuntu:~/Documents/intercept-page-fault-handler-master$ sudo ./load.sh
[sudo] password for parallels:
Sorry, try again.
[sudo] password for parallels:
parallels@ubuntu:~/Documents/intercept-page-fault-handler-master$ dmesg | tail
[ 8443.668369] Page fault Listener : page fault detected in process 29286.
[ 8443.668372] Page fault Listener : page fault detected in process 29286.
[ 8443.668380] Page fault Listener : page fault detected in process 29286.
[ 8443.668383] Page fault Listener : page fault detected in process 29286.
[ 8443.668387] Page fault Listener : page fault detected in process 29286.
[ 8443.668391] Page fault Listener : page fault detected in process 29286.
[ 8443.668394] Page fault Listener : page fault detected in process 29286.
[ 8443.668397] Page fault Listener : page fault detected in process 29286.
[ 8443.668399] Page fault Listener : page fault detected in process 29286.
[ 8443.668406] Page fault Listener : page fault detected in process 29286.
parallels@ubuntu:~/Documents/intercept-page-fault-handler-master$ ./unload.sh
parallels@ubuntu:~/Documents/intercept-page-fault-handler-master$ lsmod
Module                  Size  Used by
nls_utf8               12557  1
isofs                  39835  1
usblp                  22891  0
```

viii. Execute UserSpaceCode to receive the signal send from kernel space



```
dm_multipath           22873  0
cryptd                 20359  3 ghash_clmulni_intel,aesni_intel,ablk_helper
uvcvideo               80885  0
snd_timer              29482  2 snd_pcm,snd_seq
videobuf2_vmalloc      13216  1 uvcvideo
scsi_dh                14882  1 dm_multipath
videobuf2_memops       13362  1 videobuf2_vmalloc
snd                    69238  12 snd_ac97_codec,snd_intel8x0,snd_timer,snd_pcm,s
nd_seq,snd_rawmidi,snd_seq_device,snd_seq_midi
videobuf2_core         40664  1 uvcvideo
videodev              134688  2 uvcvideo,videobuf2_core
serio_raw              13462  0
soundcore              12680  1 snd
rfcomm                 69160  0
bnep                   19624  2
bluetooth             391196  10 bnep,rfcomm
lpc_ich                21080  0
pvpanic                12801  0
shpchp                 37032  0
prl_tg                 21944  1 prl_fs
parport_pc             32701  0
ppdev                  17671  0
lp                     17759  0
parport                42348  3 lp,ppdev,parport_pc
mac_hid                13205  0
psmouse               106678  0
ahci                   25819  3
libahci                32560  1 ahci
parallels@ubuntu:~/Documents/intercept-page-fault-handler-master$ sudo ./a.out
received value 1234
parallels@ubuntu:~/Documents/intercept-page-fault-handler-master$
```

ix. Show that intercept kernel still in kernel module



```
parallels@ubuntu: ~/Documents/intercept-page-fault-handler-master
snd_timer              29482  2 snd_pcm,snd_seq
videobuf2_vmalloc      13216  1 uvcvideo
scsi_dh                14882  1 dm_multipath
videobuf2_memops       13362  1 videobuf2_vmalloc
snd                    69238  12 snd_ac97_codec,snd_intel8x0,snd_timer,snd_pcm,s
nd_seq,snd_rawmidi,snd_seq_device,snd_seq_midi
videobuf2_core         40664  1 uvcvideo
videodev              134688  2 uvcvideo,videobuf2_core
serio_raw              13462  0
soundcore              12680  1 snd
rfcomm                 69160  0
bnep                   19624  2
bluetooth             391196  10 bnep,rfcomm
lpc_ich                21080  0
pvpanic                12801  0
shpchp                 37032  0
prl_tg                 21944  1 prl_fs
parport_pc             32701  0
ppdev                  17671  0
lp                     17759  0
parport                42348  3 lp,ppdev,parport_pc
mac_hid                13205  0
psmouse               106678  0
ahci                   25819  3
libahci                32560  1 ahci
parallels@ubuntu:~/Documents/intercept-page-fault-handler-master$ sudo ./a.out
received value 1234
parallels@ubuntu:~/Documents/intercept-page-fault-handler-master$ lsmod |grep in
tercept
intercept              13690  0
parallels@ubuntu:~/Documents/intercept-page-fault-handler-master$ 
```

Show that user process receive the kernel signal

4. Project Critical Point Explanation

There are some point worth to mention in this project. First of all, figuring out the Page Fault process. How the process begin, how the stack information store to handle the page fault exception. Then find the register which store the address we need. For example, rsp, represent stack segment in 64 bit architecture. “rdi” register represent index and pointer in 64 architecture. Secondly, figure out the page fault process and every detail in different variables, such as error code, copy-on-write characteristic in idt table, how to get function address from the global symbol table. Third, how to use assembly language to change the default page fault behavior. For our team members, it is a challenge task to write some assembly code. Since we have to very carefully deal with copy register value, also, without effect the system operation. Finally, how to send process to the specific user process. We use the “debugfs” to get the user process id, then send the message from our page fault handler. In the future, we expect to do some modification when the kernel send the signal to process, then resume to wait for user process response.

5. Conclusion

Intercept page fault in Linux kernel is not an easy task to accomplish with lack of knowledge in kernel operation. So choosing this topic is quite challenge to our team members. But in the whole project, we spend most of time to figure out the process in Linux Page fault mechanism, signal handle, and how the whole process is ongoing when the page fault exception occurs. We did gain a lot precious experience from the project. The more time we spend to research in kernel module. The more familiar with the computer architecture. Expecting the next semester, we could go deeper in the Linux kernel programming.

6. Reference

i. Daniel P Bovet,Marco Cesati. (2005). *UnderStanding the Linux kernel, 3rd edition.* O'Reily.

ii. Keller, A. (2008). *Kernel Space - User Space Interfaces.* Retrieved from http://people.ee.ethz.ch/~arkeller/linux/multi/kernel_user_space_howto-5.html

iii. LXR. (n.d.). *Linux Cross Reference.* Retrieved from http://lxr.free-electrons.com/source/drivers/infiniband/ulp/iser/iscsi_iser.c#L101.

iv. Molly, D. (2015, 4 14). *Part 1:Introduction.* Retrieved from Writing a Linux kernel Module: http://derekmolloy.ie/writing-a-linux-kernel-module-part-1-introduction/

v. Peter J Salzman, Michael Burian, and Ori Pomerantz. (n.d.). *Writing Your Own Loadable Kernel Module.* Retrieved from http://www.tldp.org/: http://www.tldp.org/HOWTO/Module-HOWTO/x839.html

vi. Phrack, K. (2003, 8 13). *Handling Interrupt Descriptor Table for Fun and Profit.* Retrieved from Handling Interrupt Descriptor Table for Fun and Profit: http://www.phrack.org

vii. Pommnitz, J. (n.d.). *Kernel level exception handling in Linux Commentary.* Retrieved from https://www.kernel.org/: http://www.kernel.org/doc/Documentation/x86/exception-tables.txt

viii. USTC, R. (2013, 04 25). *CNBlog.* Retrieved from http://www.cnblogs.com/richardustc/archive/2013/04/25/3043674.html

ix. VBird. (n.d.). *VBird WebSite.* Retrieved from V Bird: http://linux.vbird.org/linux_basic/redhat6.1/linux_06command.php#grep

x. ZHANG, S. (2011, 03 04). *May the #! with you.* Retrieved from https://onebitbug.me/2011/03/04/introducing-linux-kernel-symbols/.