Deeksha Juneja
Cpre 308
Lab 7
23/3/2016

Introduction

The purpose of this lab was to look at the Linux kernel and understand how it works. We use the linux-4.4.3 for this lab. We download the Linux kernel and then verify if it is the correct version. We run the Debian Jessie file-system and finally created the device_write module.

Questions:

List five architectures that Linux Kernel supports.

1. x86
2. ia64
3. mn103
4. powerpc
5. superh

List three file system types Linux supports

1. vfat
2. xfs
3. nfs

Output of the hello world program

Why are the prototypes not identical and and what are the extra parameters for?

```
root@myhost:/host/cpre308/labs/lab-07# cd hello-world
root@myhost:/host/cpre308/labs/lab-07/hello-world# lsmod
Module                  Size  Used by
root@myhost:/host/cpre308/labs/lab-07/hello-world# modinfo hello_world.ko
filename:       /host/cpre308/labs/lab-07/hello-world/hello_world.ko
description:    A simple hello world driver
author:         Jeramie Vens vens@iastate.edu
license:        GPL
depends:
vermagic:       4.4.3deeksha mod_unload
root@myhost:/host/cpre308/labs/lab-07/hello-world# rmmod hello_world.ko
rmmod: ERROR: Module hello_world is not currently loaded
root@myhost:/host/cpre308/labs/lab-07/hello-world# lsmod
Module                  Size  Used by
root@myhost:/host/cpre308/labs/lab-07/hello-world# modinfo hello_world.ko
filename:       /host/cpre308/labs/lab-07/hello-world/hello_world.ko
description:    A simple hello world driver
author:         Jeramie Vens vens@iastate.edu
license:        GPL
depends:
vermagic:       4.4.3deeksha mod_unload
root@myhost:/host/cpre308/labs/lab-07/hello-world# insmod hello_world.ko
Hello my amazing world!
root@myhost:/host/cpre308/labs/lab-07/hello-world# lsmod
Module                  Size  Used by
hello_world              931  0
root@myhost:/host/cpre308/labs/lab-07/hello-world# rmmod hello_world.ko
Goodbye to this boring world!
root@myhost:/host/cpre308/labs/lab-07/hello-world# lsmod
Module                  Size  Used by
root@myhost:/host/cpre308/labs/lab-07/hello-world#
```

The prototypes are not identical because the file operations call can take an offset and count if necessary. Count is needed to determine the number of bytes the program will read and the offset will offset the memory location. It is also static when the system call is not. The read system call is a standard and expects exactly three arguments.

List six operations that files can support.

1. read
2. write
3. open
4. flush
5. lock
6. poll

Expected output:
1. When the file is opened: I have already told you hello n times
2. When the file is written to: Sorry, this operation isn't supported
3. When the file is read from: No output, gets data and stores it
4. When the file is closed: No output, closes file

Output of the hello-file programming

```
root@myhost:/host/cpre308/labs/lab-07/hello-file# insmod hello_file.ko
hello_file: module license 'unspecified' taints kernel.
Disabling lock debugging due to kernel taint
I was assigned major number 254. To talk to
the driver, create a dev file with
'mknod /dev/cpre308 c 254 0'.
Try various minor numbers. Try to cat and echo to
the device file.
root@myhost:/host/cpre308/labs/lab-07/hello-file# mknod /dev/cpre308-0 c 254 0
root@myhost:/host/cpre308/labs/lab-07/hello-file# cat /dev/cpre308-0
I already told you 0 times Hello World!
root@myhost:/host/cpre308/labs/lab-07/hello-file# cat /dev/cpre308-0
I already told you 1 times Hello World!
root@myhost:/host/cpre308/labs/lab-07/hello-file# echo "hello" > /dev/cpre308-0
Sorry, this operation isn't supported.
bash: echo: write error: Invalid argument
root@myhost:/host/cpre308/labs/lab-07/hello-file# mknod /dev/cpre308-1 c 254 1
root@myhost:/host/cpre308/labs/lab-07/hello-file# cat /dev/cpre308-0
I already told you 3 times Hello World!
root@myhost:/host/cpre308/labs/lab-07/hello-file# cat /dev/cpre308-1
I already told you 4 times Hello World!
root@myhost:/host/cpre308/labs/lab-07/hello-file# rmmod hello_file.ko
root@myhost:/host/cpre308/labs/lab-07/hello-file# cat /dev/cpre308-0
cat: /dev/cpre308-0: No such device or address
root@myhost:/host/cpre308/labs/lab-07/hello-file#
```

Task for this lab

My design was inspired from the design of device_read. Basically we had to reverse the string. I was trying to do this using string.h. It wasn't really working and then I figured out that I cant use string.h. Hence, I started to explore other functions that I could use. I used get_user for reversing the string and the printk for printing it out. I am also returning the length of the string in the end. A mistake that I make was that I was printing without reversing due to the indexes that I was using. But that was a rather easy and quick fix. The output is:

```
root@myhost:/host/cpre308/labs/lab-07/hello-file-complete# mknod /dev/cpre308-0 c 254 0
mknod: '/dev/cpre308-0': File exists
root@myhost:/host/cpre308/labs/lab-07/hello-file-complete# cat /dev/cpre308-0
I already told you 0 times Hello World!
root@myhost:/host/cpre308/labs/lab-07/hello-file-complete# cat /dev/cpre308-0
I already told you 1 times Hello World!
root@myhost:/host/cpre308/labs/lab-07/hello-file-complete# echo "hello world"> /dev/pre308-0
root@myhost:/host/cpre308/labs/lab-07/hello-file-complete# echo "hello world"> /dev/cpre308-0

dlrow olleh
root@myhost:/host/cpre308/labs/lab-07/hello-file-complete# echo "this works well"> /dev/cpre308-0

llew skrow siht
root@myhost:/host/cpre308/labs/lab-07/hello-file-complete# mknod /dev/cpre308-1 c 254 1
mknod: '/dev/cpre308-1': File exists
root@myhost:/host/cpre308/labs/lab-07/hello-file-complete# cat /dev/cpre308-0
I already told you 4 times Hello World!
root@myhost:/host/cpre308/labs/lab-07/hello-file-complete# cat /dev/cpre308-1
I already told you 5 times Hello World!
root@myhost:/host/cpre308/labs/lab-07/hello-file-complete# rmmod hello_file.ko
root@myhost:/host/cpre308/labs/lab-07/hello-file-complete# cat /dev/cpre308-0
cat: /dev/cpre308-0: No such device or address
root@myhost:/host/cpre308/labs/lab-07/hello-file-complete#
```

Conclusion

Overall, this lab was very straightforward and was done rather easily. I had thought it would be connected to lab6 but I didnt really see a connection. It was interesting to use the Linux Kernel. This lab definitely helped me improve my knowledge of the Linux kernel.