Lecture 2: Installing/Reading Linux Kernel

1. What's in the linux kernel

init: start up code

 main.c : beginning of linux

kernel: cpu-indendent code for basic system management

 fork.c, exit.c, sched.c : process creation, exit, scheduling

arch : cpu-dependent code for basic system management

 arch/x86 : cpu-dependent code for intel x86 cpu

 arch/x86/boot : boot-related code

 arch/x86/kernel: low-level intel cpu code for initialization, interrupt handling, etc

 entry\_32.S : code for interrupt entry point

 head\_32.S : code for system initialization

 i8259\_32.c : code for interrupt controller

 irq\_32.c : code for IRQ

 process\_32.c : code for process control

 time\_32.c : code for timer

 traps\_32.c : code for exception handling

 signal\_32.c : code for signal handling

fs: file system code

 open.c, read\_write.c, file.c, ... : fs system call handling

 inode.c : inode handling

 fs/ext2 : ext2 file system code

 fs/ntfs : windows NT file system code

mm : memory management code

net: network handling code

drivers: device handling code

include : include files

 inclde/asm-x86 : include files for intel cpu specific code

ipc: code for inter-process communication

2. Linux starting location: init/main.c: start\_kernel()

3. homework

1) Install Gentoo Linux on virtual machine.

1.0) Download Virtualbox from Internet and install.

1.1) Download Gentoo virtualbox files (gentoo.zip) from the I-class and un-compress it.

1.2) Run VirtualBox, and click File>Import and go to the gentoo directory. Select Gentoo2.ovf. Uncheck USB controller. Select Import. This will install Gentoo Linux on your virtual box.

1.3) Run Gentoo. If you have an error, run VirutalBox as administrator and try again. For USB error, simply disable usb controller in "Setting" tab. For Hyper-V error (Raw-mode is unavailable), turn off Hyper-V feature in control panel>program and feature>window feature. Select My Linux. Login as root and hit Enter for the password prompt. If VirtualBox still cannot open the session, you need to look at the log file (right click on Gentoo VM and select “log file”) and see what is the error and fix it.

(In some cases, you may need to download and install virtualbox extension package.)

1.4) Make a simple program, ex1.c, with vi which displays "hello world". Compile and run it.

2) Go to linux-2.6.25.10 directory and find all the files referred in Section 1 such as main.c, fork.c, entry\_32.S, etc.

3) Find the location of start\_kernel().

 To find a string "start\_kernel", go to the linux top directory (linux-2.6.25.10) and do

 $ grep -nr "start\_kernel" \* | more

 Use "space" to move down the screen, "q" to exit

 Once you found the file that has "start\_kernel", use "vi" to read the file.

 In vi, type "/start\_kernel" to search for the first instance of "start\_kernel".

 For the next string, simply type "/". Repeat "/" until you find the start\_kernel() function.

 Use "j" to mode down the cursor, "k" to move up, "^f" to move one screen down, "^b" to move

 one screen up.

4) start\_kernel() is the first C function run by Linux. Predict what will be the first message appearing on the screen by analyzing start\_kernel(). Note that "printk()" (not "printf") is the function to print something in the kernel. Confirm your prediction with "dmesg > x" and "vi x". The kernel remembers the booting message in the system buffer and dmesg displays the content of this buffer to the screen. "dmesg > x" will send the booting message to file x. With "vi x" you can look at the file x.

5) Find the location of the following functions called in start\_kernel() and briefly explain your guessing about what each function is doing (but do not copy the whole code).

 trap\_init(), init\_IRQ(), sched\_init(), time\_init(), console\_init(), mem\_init(),

 rest\_init()

6) Why can't we use "printf" instead of "printk" in Linux kernel?