Project 0:–
Linux and Virtual Machine Dabbling

Professor Hugh C. Lauer
Professor Robert Walls

CS-3013 — Operating Systems

(Slides include copyright materials from *Operating Systems: Three Easy Step*, by Remzi and Andrea Arpaci-Dusseau, from *Modern Operating Systems*, by Andrew S. Tanenbaum, 3rd edition, and from other sources)

In this project, we will …

- **Install our virtual machines and learn how to use them**
  - This is the place you will work during this course!

- **Modify, build, install, and test the Linux kernel**
  - With your name on it!

- **Create and test a *Loadable Kernel Module* (LKM)**

- **Turn in the project using *InstructAssist***

In this project, we won't …

- **… … try to support more than one machine architecture**
  - However, 32-bit and 64-bit architectures are different at the kernel level!
  - Hardware-level code for different processor families is organized differently in kernel source code!
  - Your kernel is not portable from one to another

Using your Virtual Machine

- **Guest OS**
  - *Ubuntu 16.04.3*
  - Configured for projects of this course
  - Configured to work with *VirtualBox*
  - Can be used with other virtualization platforms – e.g, *VMware*
  - Basic devices needed for this course
- **Host system**
  - Your own or corporate PC or Mac
    - *Virtual Box* application installed
  - Zoo Lab — See professor
    - *Virtual Box* with VM on flash drive
  - *Parallels* — we will learn about together!

Virtual Machine on your own computer

- **Download and extract from**

  http://cs.wpi.edu/~cs3013/c18/Resources/CS-3013_Virtual_Machine.ova

- **See "cookbook" for how to clone or copy**

  http://cs.wpi.edu/~cs3013/c18/Resources/SettingUpYourVirtualMachine.docx
  http://cs.wpi.edu/~cs3013/c18/Resources/SettingUpYourVirtualMachine.pdf

- **Open in *VirtualBox***
  - Adjust properties per cookbook

See Tools > Assignment Instructions on IA

- **Use separate virtual machine for this course!**
- **Don't waste too much time trying to figure it out on your own**

RTFM and/or get help!

Starting your Virtual Machine

- **Start your virtual machine**
  - Login as `student`, password = `C-Term18`
- **Reinstall "Guest Additions"**
  - VirtualBox tools to move between host and guest
  - Host specific
- **Switching between Host and Guest**
  - Click in guest window to focus mouse and keyboard
- **Full screen mode**
  - Use Virtual Box menu (popup at the bottom)
- **Interrupting**
  - `CTL-ALT-DEL` *always* goes to host system
  - Use *VBox* menu command Input > Keyboard > Insert-Ctrl-Alt-Del

Ubuntu Desktop

- **Looks / acts somewhat like Windows or Mac GUI**
- **Many similar tools and applications**
- **Toolbar on left has common applications**
- **To open command shell — *CTRL+ALT+t***
- **Create new user identity for yourself**
  - Click on “WPI Student” in upper right “gear” menu
  - Make sure your new account is of *Administrator* type

Other Notes

- **When input focus is in guest window**
  - Inserted CD/DVD is recognized by guest OS
  - Same for USB flash drive
- **Adjust processor settings**
  - Amount of virtual RAM
    - Suggest half of host RAM
  - Number of virtual processors
    - Suggest half of host processors

Usually!

If not, see VM or Virtual Machine menu

When VM power is off!

Share your thoughts and experiences with your colleagues

Use the Forum on InstructAssist!

Suggestion

- **Archive your virtual machine now**
- **So you have a preserved copy for future projects!**

- **You will be using it throughout the course**
- **You will probably mess it up!**
  - At least once, perhaps more than once!

Questions?

Part 1: Building the Linux Kernel

- **Follow the instructions in Project 0 description**
- **Download sources via `git clone`**
  - Takes 5-8 minutes on campus
- **Three "`make config`" steps**
  - Selects modules to build
  - Restricted to features/facilities actually in use
  - Opportunity to *add your name* or other tag
- **`make –j`*`n`***
  - Compiles using *n* processors
  - 5-7 minutes on medium Core i7 using 4 processors

Part 1: Installation and demonstration

- **`sudo make modules_install install`**
- **Reboot.**
- **Press and hold Left-Shift key *as soon* as VirtualBox splash screen appears**
  - Release when boot menu appears
  - Select your kernel (with your name!)
  - Demonstrate with “**`uname -r`**”

Questions?

Part 2:– Loadable Kernel Module

- **What is a Loadable Kernel Module (LKM)?**
  - Reference *Linux Kernel Development,* Chapter 17
- ***Answer:* a module that can be loaded into the kernel *at run time!***
  - Written by you!
  - Requires administrative permissions to load and unload
- **This project:–**
  - Copy the LKM code from project specification
  - LKM writes to system log upon loading & unloading
  - Test by listing tail of system log

Part 2a: make a LKM

- **Code provided in Project 0 document**
- **Read and understand**
- **Compile in user space**
  - And then add to kernel via **`sudo insmod`**
  - Verify in **`/var/log/syslog`**
  - Remove via **`sudo rmmod`**
  - Verify again in **`/var/log/syslog`**

Part 2b:– Modify Linux kernel source

- **Use same sources previously downloaded to build kernel**

- **Add three tiny system calls**
  - Code is provided in the handout
  - Makes entries in system log

Discussion: Why do we have system calls?

- **What is a system call?**
  - A function in the kernel; runs in *privileged* mode
  - Invoked by special assembly language instruction
  - Kernel indexes into a table to find right function

Part 2: Rebuild the Modified Kernel

- **Configure kernel**
  - Add your name to *Local Version* setting
    - Lower case only, and +, –, 0-9, periods
  - **`make menuconfig`**
- **Recompile kernel**
  - Only updates subset of modules
- **Install kernel**
  - **`sudo make modules_install install`**

Part 3: Test and submit modified kernel

- Code provided in project description
- Results written to `/var/log/syslog`
- Submit per instructions in Project document
  - To `InstructAssist`

Questions?