The Linux Kernel

System Administration Decal
Lecture #5
Joshua Kwan

How was the lab?

- Did you understand SSH key authentication? You should use it yourselves!
- Did you manage to annoy the hell out of each other using 'write' and 'wall'?
- Did you figure out how to get NetHack to build? Did you play?

Today

- What's the structure of the Linux kernel?
- Who codes on the kernel?
- How do you deal with loading drivers into the Linux kernel while it's running?
- How do you build the Linux kernel and run it on your system?

Linux Kernel Modules

- Kernel modules are versioned chunks of code that can be loaded directly into the kernel to provide additional functionality
- ~90% of hardware drivers can be built as modules, others have to be built-in.
- Commands: **modprobe, insmod, rmmod, modinfo, lsmod**

Linux Kernel Modules

- **modprobe**: Loads a module and its dependencies (must run **depmod** first)
- **depmod:** Calculates module dependencies (e.g. SCSI controller driver needs SCSI base code)
- **insmod, rmmod:** Inserts and removes modules without dependency checking
- **lsmod**: Lists currently loaded modules

The Linux Kernel

- Like any other free software, kernel source is just a bunch of C files and Makefiles
- Differences:
  - Elaborate configuration
  - Build process generates a bootable file as well as external modules.
- Download the source at www.kernel.org

Source Directories

- `arch`: code that is architecture-specific
- `fs`: filesystem drivers (Ext2, XFS, ReiserFS)
- `drivers`: Hardware drivers for everything under the sun (video, network cards, etc.)
- `sound`: For some reason, sound card drivers are in here and not in drivers/.

Who does all this stuff?

- Linus Torvalds, original creator, essentially supervises code checkins at this point
- People generally specialize in a small area of the kernel: SATA, sound, task scheduling, memory management, drivers…
- Mandatory review and approval process; a very organized system

Configuring the Kernel

- “make menuconfig” demo!
- Note the difference between building things into the kernel image and building modules.
- Most Linux distributions build the smallest kernel possible, and ship all possible modules

Building and Installing

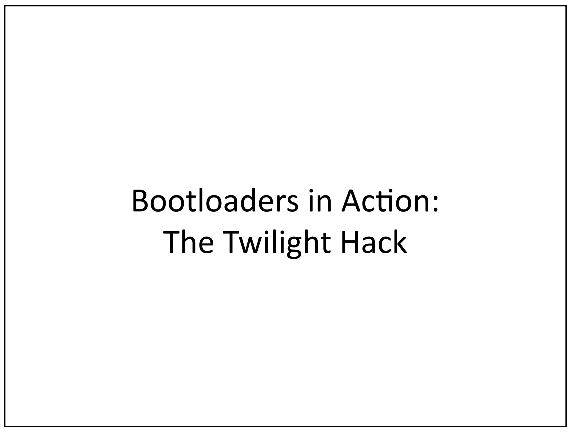
- A simple ‘make’ builds the kernel image and all requested modules.
- ‘make install’ will install the kernel image and ‘make modules_install’ will install modules.
- You might have to update your bootloader, such as GRUB or LILO.

What’s a bootloader?

- A chunk of code that runs at the BIOS level, loads the kernel into memory and executes it.
- Hard problem
  - Must have small footprint
  - Needs a filesystem driver
  - Read the config file
  - Use BIOS calls to access the HD and execute the kernel.

GRUB and LILO

- Using your virtual servers, you won’t have to worry about booting Linux off bare hardware.
- But do install Linux and home and check it out!
- **LILO**: legacy, monolithic bootloader (must re-install boot sector every time you update the kernel)
- **GRUB**: modular bootloader that reads config file off the disk and has a shell. You can have a hosed config and still manage to boot your system.



Bootloaders in Action: The Twilight Hack