Introduction to Linux

History

- MINIX was created in 1987 by Andrew Tanenbaum as a minimalist Unix for students
- Linus Torvalds forked Linix off of this in 1991
  - Frustrated with educational-only licensing of MINIX
  - Wanted to bring Unix to the popular x86 processors
- Is now at version 4.3
  - Updates weekly

Directory Structure

- /
  - /boot for boot loader and kernel binaries
  - /dev hardware devices
  - /sys vfs to access kernel data structures
  - /proc vfs to access kernel processes
  - /bin minimal set of binaries to run
  - /sbin /bin but requires root
  - /lib shared libraries (similar to DLLs)

Directory Structure

- /
  - /media common mount point for external devices
  - /mnt temporary mount point for external devices
  - /etc configuration files
  - /usr ...
  - /home user documents
  - /root /home for root user
  - /var persistent access to temporary files
  - /tmp fast access to temporary files

Directory Structure

- /usr
  - /usr/bin — most system binaries
  - /usr/sbin — /usr/bin for root access
  - /usr/include — header files
  - /usr/lib — program libaries
  - /usr/local — self-compiled binaries
  - /usr/share — documents, icons, etc
  - /usr/src — kernel source trees

What to Use

- Lab machines in MCH 202
  - Email me group layout by Sept 30th
    - Exactly 3 people per group
    - I can also help you find partners until the cutoff
  - After the cutoff, I will start randomly assigning groups
    - This is to ensure every group has a machine
    - 65 students / 21 machines
  - I will then assign you a username, password, and machine
    - You must implement Project 2 on that machine
- For project, use kernel version 4.2
  - kernel.org/pub/linux/kernel/v4.x/linux-4.2.tar.xz

Initial Setup

$ sudo apt-get install libncurses5-dev

Downloading the Kernel

```
$ mkdir ~/test_kernel/
$ cd ~/test_kernel/
$ wget kernel.org/pub/linux/kernel/v4.x/linux-4.2.tar.xz
$ tar Jxvf linux-4.2.tar.xz
$ cp -R linux-4.2/* ./
$ rm -Rf linux-4.2
$ rm linux-4.2.tar.xz
$ cd ~/
$ sudo mv ~/test_kernel /usr/src/
$ cd /usr/src/test_kernel
```

Compiling the Kernel

$ make menuconfig

- Graphical configuration setup
- Stored in .config

$ make

- Compiles source tree

$ make modules_install

- Installs module binaries into modules/

$ make install

- Installs binaries into /boot

$ sudo shutdown -r 0

make menuconfig

- Goal is to remove as much as possible without making it unbootable
  - Reduces the resulting binary and decreases boot time
- Each item has a tristate
  - [*] Installed in the kernel directly
  - [M] Installed as a kernel module
  - [ ] Not installed at all

- Need to remove anything won't be used
  - *lspci* to view hardware
- Module candidates are things that you may need later but don't warrant loading every time
  - You'll probably have very few of these as you're doing debugging on a static environment
- Include everything else directly in the kernel

- If overwhelmed, just use an old, working configuration
  - I'm not grading your ability to install a stripped down kernel

make oldconfig

- Uses old configuration to build a new one

  $ sudo cp /boot/config-3.16.0-38-generic /usr/src/test_kernel/.config

  $ sudo make oldconfig
- Accept changes, then you can use *make menuconfig* to edit this down
- This will get you a working setup in case things go wrong

Booting Problems

- What would happen if you set all the disk drivers as modules...
- You wouldn't be able to boot into your kernel
- This is because
  - The boot loader loads the kernel image from /boot
  - The kernel then takes over, but doesn't know how to find /
- Solutions
  - Try each disk driver one by one until you can't boot
    - Time consuming initially
  - Include them all
    - Wasteful (time consuming when loading)
  - Use initramfs

initramfs

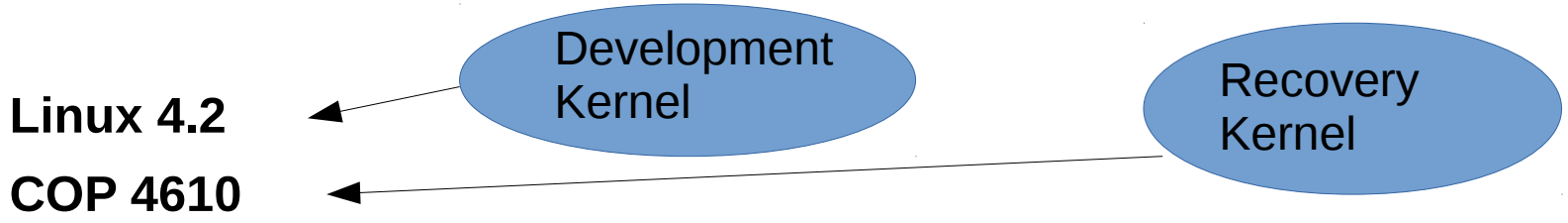
- Creates image file that automatically loads modules needed for boot

```
$ cd /boot
$ sudo mkinitramfs -o initrd.img-4.2.0 4.2.0
```

When it Doesn't Boot

- Load original kernel
  - You should always have at least one working kernel
- Check that you didn't skip any steps
- Try adding some features back in
  - Use make oldconfig if things get really bad
  - Add things one at a time

Lab Machines



Advanced options for Linux Mint 17.2 Cinnamon 64-bit

- Linux Mint 17.2 Cinnamon 640-bit, with Linux 3.16.0-38-generic
- Linux Mint 17.2 Cinnamon 640-bit, with Linux 3.16.0-38-generic (recovery mode)

CIS 4930

Advanced options for Linux Mint 17.2 Rafaela (17.2) (on /dev/sda5)

- Linux Mint 17.2 Cinnamon 64-bit (on /dev/sda5)
- Linux Mint 17.2 Cinnamon 64-bit, with Linux 3.16.0-38-generic (on /dev/sda5)
- Linux Mint 17.2 Cinnamon 64-bit, with Linux 3.16.0-38-generic (recovery mode) (on /dev/sda5)

Memory test (memtest86+)

Memory test (memtest86+, serial console 115200)