Ubuntu Installation

Don Porter

Overview

- Similar to Windows, but more complicated
- Note “Test Memory” on boot screen (memtest86)

- A lot of the dialog boxes are based on ASCII (using a library called ncurses)
  – Use Tab to cycle between options

Detect Keyboard layout?

- I usually skip this and use the default PC keyboard

Hostname

- This is generally the (short) name you want to appear on the network
- So if your machine is going to be named kermit.oscar.cs.stonybrook.edu set the hostname to kermit
  - You can change this later by editing /etc/hostname and /etc/hosts

Partitioning

- I usually use guided without LVM
  - LVM makes it easier to add disks later, but also adds some administrative complexity
  - Probably good software
- Default sets aside a little space for swap, makes the rest ext4
  - Ext4 is a fairly standard file system format
  - Swap space is a region of disk dedicated for spilling memory to under pressure

Automatic Updates

- I tend to only do security updates
- On a development system, library updates can sometimes break applications, making regressions hard to debug
  - I find it better to do this once a week on a known day

Server Software

- Ubuntu is nice and offers to set up some common software packages automatically
- Let’s just install openssh for remote access

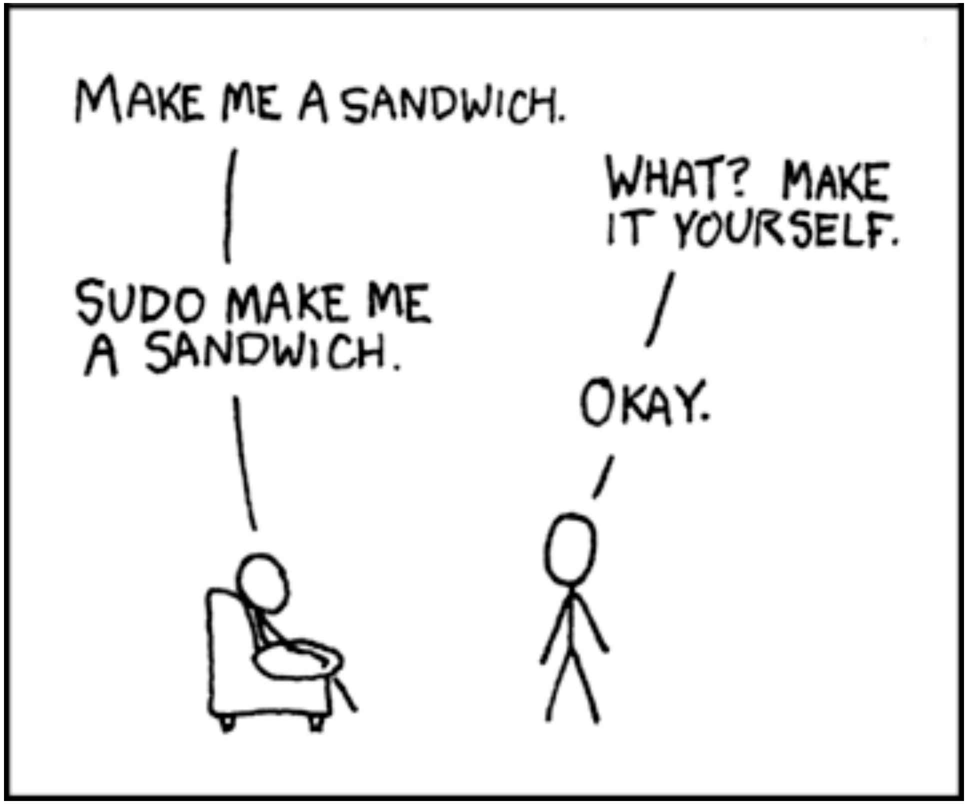
GRUB

- GRand Unified Bootloader
- Basically, the thing that runs between the BIOS and the OS kernel
  - GRUB is a popular bootloader for Linux
  - Fixed some deficiencies in the previous LInux LOader (LILO)
  - Capable of booting about anything, including Windows
  - I prefer Grub version 1. Grub 2 is sort of unwieldy
- What is the master boot record? The region of the disk that is read by the BIOS to find the bootloader. There can only be one.

sudo

- Good practice not to always run as Administrator
  - root in Linux parlance
  - You might accidentally type ‘rm –rf /’
- Log in as yourself, type ‘sudo <command>’ to issue just that command as root

Fun from xkcd



Software installation, the bad old way

- Most Unix/Linux software is distributed as source
  - Installation involves both compilation and copying binaries to common directories (e.g., /usr/bin)
- Most common approach: make
  - At the command line:
    - make
    - sudo make install

The problem with make

- No easy way to enumerate what high-level software is installed
- Hard to completely uninstall a package
  - Especially if you lose the original makefile
- Ad hoc (or no) mechanisms to track dependencies
  - E.g., emacs uses many libraries, may depend on specific versions
- Hard to centralize and automate security patching

Package managers

- Software installation systems
  - Track files installed, dependencies
  - Centralize software distribution and updates
  - Takes a little extra work to convert makefiles into packages
- Common ones:
  - Ubuntu/Debian
    - Debian package manager (.deb) for individual packages
    - Distributed using advanced package tool (apt)
  - RedHat/Fedora/Centos
    - Redhat Package Manager (.rpm) for individual packages
    - Distributed using yum

apt

- sudo apt-get update
  - Pull just the list of available updates
- sudo apt-get upgrade
  - Update most packages. Holds back a few sensitive ones, like the kernl
- sudo apt-get dist-upgrade
  - Update everything

apt, cont

- apt-get install foo
  - Install the foo package
- apt-get remove foo
  - Uninstall foo
- apt-get remove --purge foo
  - Uninstall foo and delete any configuration files
- apt-get autoremove
  - Delete any dependent packages that aren't needed anymore

apt-file

- Creates a database of file-to-package mappings
- Useful to figure out what package you need to install for a given binary, library, header file, etc.
- apt-file update
  - Download the package information
- apt-file search libc.so
  - List all packages that install a file matching pattern libc.so

Denyhosts

- A good idea to install
- Basically, this black-lists any hosts/IP addresses that try to log into the server and fail more than 5 times
  - Prevents brute force password guessing attacks
- Configurable
  - Defaults are a bit draconian. You really want to rest the count after a successful login

Editing configuration files

- Often, only vi is installed
- I prefer emacs. Easy enough to install
- Worth learning one or the other
- A few command cheat sheets are easily found online

Custom Linux kernel build

- Linux does a lot of its configuration at compile time
  - Avoid littering the code with needless branches
- Problems:
  - An option you need (e.g., a driver) may be compiled out
  - You may have a bunch of code enabled you don’t want
- Useful to know how to compile a custom Linux kernel

Download

- Use an FTP client to download the kernel source from ftp.kernel.org
  - I recommend ncftp
- Unpack using ‘tar –jxvf’
- cd linux-X.X.X

Configuration

- Stored in a file called .config
  - Also in /boot/config-XXX
  - Can copy /boot/config-XXX to .config to adopt current config
- make menuconfig – ncurses-based interface
  - make oldconfig checks an existing .config

Compilation

- make works
- I use make-kpkg to get a .deb instead
  - make-kpkg linux_image --rootcmd fakeroot --initrd --append-to-version=<your custom version name>
  - Then dpkg –i linux-image-XXX.deb
- And select from grub during boot

Questions?