**Linux**

**Agenda**

* Intro to Linux
* Demo installation
* Look at GUI, command line

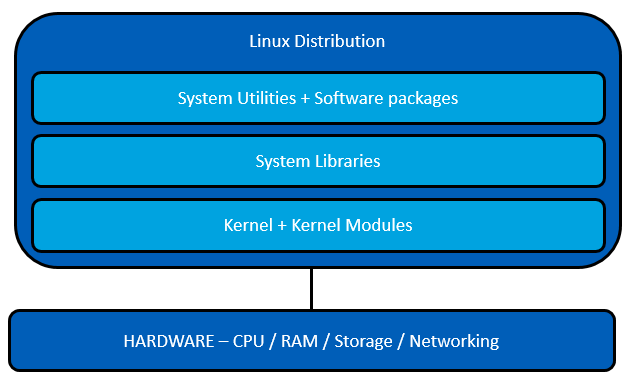
**Introduction**

**Why Linux**

* It’s everywhere
* Mainly server-side OS
* Minimal/no licensing costs
* Workstations
* Embedded appliances (networking hardware, Internet of Things)
* Mac OS/X and iOS (UNIX)
* Android

**Introduction to Linux – Linux Components**

* A Linux distribution has 3 primary components:
  + Kernel
    - This IS Linux
  + System Library
  + System Utilities



**Linux Timeline**

* 1986 – UNIX – AT&T Bell Labs
* 1977 – BSD – UC Berkeley
* 1979 – First commercial GUI (made by Xerox)
* 1983 – Free Software Movement
  + The idea that code should be shared, not that code should be free
  + No need to reinvent the wheel
* 1984 – Apple Macintosh
  + Ran a version of BSD
* 1985 – Microsoft Windows 1.0
* 1991 – Linux created
* 1993 – Debian released
  + First fully-functional open-source Linux distribution
* 1994 – Red Hat released
  + Large company today
  + Open-source; makes money from licensing
* 2004 – Ubuntu released
  + Made to be user-friendly

**A Short History of Everything \*nix**

* The “Unixes”
  + HP-UX
  + AIX
  + Solaris
  + BSD variants
    - macOS/iOS

**Today’s Distribution “Tree”**

* <http://i.stack.imgur.com/NZx8g.png>
* Google “Linux timeline” for more
* Debian, S.u.S.E (pronounced “Sue say”), and Red Hat are the main branches
* Debian uses a tool called APT for package delivery
* Red Hat used Yum, uses something else now
* APT is better, but both are fully-functional now
* S.u.S.E is a commercial variant of Slackware
  + Now default Microsoft deployment option
* S.u.S.E and Red Hat are paid-for
* Red Hat has annual licensing fees, more expensive than Windows
* Enterprise environments tend to run Red Hat, development tends to run Debian
* Debian development, then “roll” it into Red Hat for enterprise

**Open Source Software**

* 1983 – Free Software Movement – Think free speech, NOT free lunch
* 1998 – “Free Software” term replaced with “Open Source Software”
* GNU = GNU’s Not Unix; this is a recursive joke by geeks <https://www.gnu.org/home.en.html>
* The value of open source include:
  + Security
    - Because of complexity; extremely large values that are hard to crack by brute force
    - Lots of oversight/”eyes on the software” to look for algorithm flaws
  + Affordability
  + Flexibility

**Class Exercise – Learn About Linux**

* Topics:
  + GNU/Linux
  + Windows vs Linux
  + UNIX vs Linux
  + History of Linux
  + Debian vs Red Hat

**History of Linux**

Linux is based on UNIX which was created by Linus Torvalds. The original Linux kernel was forked into all subsequent versions. Any time you see a new branch on the distribution tree is where basically someone disagreed with the current way of doing things and forked away.

UNIX was originally developed for a multi-user OS by MIT, AT&T and GE. The dominant GUI that was created influenced everything that came out after it. After AT&T dropped out of the Multics project, UNIX was conceived and implemented by Ken Thompson and Dennis Ritchie. They later rewrote it in C (new at the time), to make it portable.

BSD was developed in 1978 by the Computer Systems Research Group from UC Berkeley, based on UNIX 6th edition. AT&T sued them since BSD contained UNIX code that they owned, hampering the development and adoption of BSD.

In 1983, Richard Stallman started the GNU project to create a free UNIX-like OS. The GNU kernel, called Hurd, failed to attract enough development effort, so GNU was incomplete. This was a factor that led to Torvalds to start his project.

In 1991, while studying Computer Science at University of Helsinki, Linus Torvalds started the Linux kernel, on the GNU C Compiler. He eventually wrote an OS kernel. On August 25, 1991, he (at age 21) announced Linux in a Usenet posting. He wanted to call this Freax, a portmanteau of free, freak, and x (an allusion to UNIX). He did consider the name Linux, but initially dismissed it as too egotistical. A colleague of his, Ari Lemmke at Helsinki University of Technology, he didn’t like Freax, so he named the project Linux on the server without consulting Torvalds, who would eventually consent to the new name.

Torvalds first published the Linux kernel under its own license, which had a restriction on commercial use.

Torvalds announced in 1996 that there would be a mascot for Linux, a penguin, because when they were about to select one, Torvalds mentioned he was bitten by a little penguin previously.

**Certification Options**

Red Hat:

* <https://www.redhat.com/en/services/certification>
* Red Hat Certified System Administrator
* Red Hat Certified Engineer
* Advanced level certifications

Linux Professional Institute:

* [https://www.lpi.org](https://www.lpi.org/)
* Linux Essentials
* LPIC-1 to 3

**Image Distribution**

* Distribution “distros” websites
* Mirror sites
* Usually ISOs
* <http://www.redhat.com/en>
* <https://www.centos.org/>
* <https://getfedora.org/>
* <https://www.debian.org/>
* <http://www.ubuntu.com/>

**Installation**

**Hardware Requirements**

Red Hat Enterprise Linux v7 Recommended Minimum Hardware:

Minimum Memory:

* 32 bit CUP – 1GB memory/logical CPU (512MB minimum) – v6 only
* 64 bit CPU – 1GB memory/logical CPU

Minimum Disk Space:

* Should be above 10GB, recommend 20GB minimum

Installation Media:

* Optical drive, USB port, ISO file mounted to VM, network connectivity for network install, etc.

**Linux File System**

* Physical vs Logical
* LVM – Logical Volume Manager
* Common mountpoints – why split things up?
  + / (root partition)
  + /boot (approx. 1 GB)
  + /lib
  + /bin
  + /sbin
  + /opt
  + /usr
  + /tmp

**Disk Partitioning**

* Most OSs can be installed on a single, un-partitioned hard disk
* Partitions offer some important advantages:
  + Ease of use
  + Performance
  + Security
  + Backup and Recover
  + Stability and Efficiency
  + Testing

**Swap Space**

* Windows = “Virtual Memory” or “Page File”, kept as file on disk
* Linux = separate disk partition
* Swap allows for moving memory pages between RAM and disk
* When device is running out of RAM, swap is used as temporary memory
* Unused items in RAM are “swapped” into swap memory

|  |  |  |
| --- | --- | --- |
| **RAM in System** | **Recommended SWAP** | **If Hibernation Required** |
| <= 2GB | 2 \* RAM | 3 \* RAM |
| >2GB – 8GB | 1 \* RAM | 2 \* RAM |
| >8GB – 64GB | At least 4GB | 1.5 \* RAM |
| >64GB | At least 4GB | Not recommended |

**Disk Partitions**

File systems MUST HAVE their own partitions:

|  |  |
| --- | --- |
| **Partition** | **Purpose** |
| / | PPT slides for this table were wrong… |
| /boot |  |
| swap |  |

File systems that often have their own partitions (especially last 3):

|  |  |
| --- | --- |
| **Partition** | **Purpose** |
| /usr | Executable binaries, kernel source tree and most documentation |
| /var | Variable sized files. Spool directories for mail and printing go. Error log directory. Application files (web server) |
| /tmp | Temporary data files, full read and write for all users |
| /home | Users home directories |

**Linux Installation**

* Without a well thought-out partition schema, the following attacks can take place:
  + Runaway processes
  + Denial of Service attack against disk space
  + Users can download or compile programs in /tmp or even in /home
  + Performance tuning is not possible
  + Mounting /usr as read only not possible to improve security
* Each partition can be mounted with different options to prevent these issues

**Users**

* Root
  + Administrator user for system
  + Unsuitable for normal use
* Admin
  + Part of “Sudo” or “wheel” group, has root permissions (full administrative access)
  + When logged in as a regular user, type command su, then hit enter to switch to root, enter root password, then do administrative action(s)
  + Another way, in command prompt, type sudo <command to run as an admin> this will run the command as the root user, prompts for MY password (not root password). Will let you do admin actions if you have permission for this. Better than letting anyone use full root permissions (very few people should have that)
* Regular users
* Changing users and permission levels:
  + Su (switch user)
  + Sudo (switch user and do command)

**Packages**

* Now vs Later
* For GUI, select it during installation (now)
* If you know you need it, check the box. If you don’t know, don’t select it (makes installation bigger)

Installation Notes for Virtual Machine – Final Exam

* Open VMWare
* Create a New Virtual Machine
* Choose Custom (advanced)
* Hardware compatibility (for using older software). In this class, we can go back to Workstation 8, but we’re going with 15
* Choose “Installer disc image file” and then browse to ISO we downloaded, it will detect the OS type (CentOS 7 for us)
* Make virtual machine name, set location if desired
* Processor Config: For this class: number of processors: 1 with 2 cores per processor
* Memory for the virtual machine: 4096 MB (4 GB)
* Network Type (can be changed later): Use bridged networking
* Select I/O controller: default
* Disk Type: SCSI (default)
* Select a Disk: Create a new disk
* Specify Disk Capacity: Max disk size: 30 GB; store virtual disk as a single file (multiple files for FAT32 disks, like some USB drives)
* Next screen: default
* Ready to Create Virtual Machine: confirm settings, uncheck auto power-on, then finish

After powering on the VM for the first time:

* You’ll lose mouse (move with arrow keys)
* Test the media (follow the prompts), then install CentOS 7
* Installer is called Anaconda
* Can pick both keyboard language (for layout) and reading language
* Next screen will be for setting up things like date and time, language support, etc.
  + Software selection: Default is minimal install with no add-ons; this will be command-line only (no GUI). We’ll use Server with GUI option (first option to include a GUI). We’ll use no add-ons
* Can ignore KDUMP and SECURITY POLICY; select “network & host name” and turn it on, then click done
* Device Selection: Change “Partitioning” to “I will configure partitioning” from automatic, click Done at top to open Manual Partitioning tool
  + Make sure “LVM” is selected
  + Click create partitions automatically
    - This will give you some partitions that the automatic option above would’ve made
    - We’ll reduce the size of the root partition (look at the lab doc – 10 GB)
    - Make the change and click somewhere else to commit the change; you’ll be notified that you have ~16GB of free space now
    - Change size of other partitions as required
    - Need to create /opt and /home
      * Click “+” (Add), select mount point /home (/opt isn’t there), can change desired capacity
      * For /opt, need to type in manually
      * **Remember: Linux is case-sensitive**
      * File system type for /home is ext4, so select /home space, and can change “file system” to ext4 (under Device Type, which is LVM)
      * Click done, will produce a summary of changes, confirm it then click done
* Click “Begin Installation”
* Set up first users:
  + Root password
  + Full name for user (which is automatically username)
    - For an admin user, check “Make this user administrator”
    - Enter password
* Wait for installation to finish

**Boot**

**Booting**

* Booting process:
  + POST/MBR
  + GRUB (**GR**and **U**nified **B**ootloader)
  + Kernel (/sbin/init)
  + Init… Runlevel
* All boot activities are recorded to /var/log/boot – great file for troubleshooting

**Runlevels**

1. Power Down -> poweroff.target
2. Single User Mode -> rescue.target
3. Multiuser, no network/NFS -> multi-user.target
4. Multiuser with network -> multi-user.target
5. Not used -> multi-user.target
6. Graphical mode -> graphical.target
7. Restart -> reboot.target

**Command Line 101**

* Right-click -> “Open in Terminal”
* Will be different on other distributions
* Case sensitive!
* Spaces = AWFUL
* Special characters to avoid: ~, <, >, /, \, $, &, :, ;

**GUI**

**Desktop Environment**

* X Windows – AKA X11 – X.org
  + GUI Framework – Windowing System
  + Architecture independent
  + Network protocol built into service
  + Designed to be used over network connections
  + Client/server relationship is “reversed”
    - Server accepts graphical output (Windows) from client
    - Server sends user input (keyboard/mouse/touch) to the client

**Window Managers**

* Controls placement and appearance of Windows within the GUI
* No additional software included
* Common X Window Managers:
  + Compiz
  + Kwin
  + Metacity
  + Enlightenment
  + MANY others…

**Desktop Environments**

* GNOME:
  + Desktop environment
  + Window Manager = Mutter
  + Includes extensive software packages
  + Free and Open Source
  + Part of GNU Project
  + Wide distribution
    - Red Hat/Fedora/CentOS
    - Debian/Ubuntu/Kali
    - Oracle Linux
    - SteamOS

**GNOME Alternatives**

* Cinnamon
  + Based on GNOME, distributed with Linux Mint
* KDE
  + Free desktop environment
  + Broad support from development community
  + KDE Plasma – current Workspace based desktop environment
  + MANY KDE developed utilities and tools
  + Wide distribution support
    - Arch Linux/Debian/Kubuntu/Linux Mintf
    - FreeBSD/Gentoo/OpenSUSE
    - Windows!
* UNIX Desktop Environments
  + CDE – HP/IBM/Solaris/USL/OpenVMS
  + VUE – HP-UX
  + IRIX – Silicon Graphics (SGI)