澳 門 理 工 學 院
Instituto Politécnico de Macau
Macao Polytechnic Institute

COMP 225: Network and System Administration Notes #1: About Linux

K. L. Eddie Law, PhD

Associate Professor

Computing, MPI

Academic Year 2nd Semester, 2019-2020

Topics

- Introduction to Unix
- History of Unix
- What is Linux?
  - OS structure
  - File system
- Linux distributions
- The shell
- Users, groups, and permissions

X

Introduction to Unix

- Unix is a multi-user, multi-tasking operating system
- Many users may log into a system simultaneously, each running many programs
- The kernel (OS) is responsible for keeping processes and users separated and regulating accesses to system resources, e.g., hardware (e.g., CPU), memory, disk and other I/O devices

---

History of Unix

- First Version was created in AT&T Bell Labs in 1969
  - Bell Labs were in Lucent after splitting from AT&T
  - Lucent was acquired by Alcatel during tech bubble in early 2000's
- Some famous Bell Labs programmers worked on this project
  - Ken Thompson, Dennis Ritchie, Rudd Canaday, and Doug McIlroy designed and implemented the first version of the Unix File System on a PDP-7 along with a few utilities
  - The name Unix was given by Brian Kernighan

History of Unix (cont'd)

- 00:00:00 Hours, Jan 1, 1970 was the time zero for Unix, also called as an *epoch*
- 1973 Unix was re-written mostly in C, a new language developed at that time by Dennis Ritchie
- Being written in this high-level language (comparing to machine languages) greatly decreased the effort needed to port it to new machines

5

Evolution

- 1977: There were about 500 Unix sites world-wide
- 1980: BSD 4.1 (Berkeley Software Development) at UCB
- 1983: SunOS, BSD 4.2, System V
- 1988: AT&T and Sun Microsystems jointly develop System V Release 4 (SVR4), which was later developed into UnixWare and Solaris 2
- 1991: Linux was created by Linus Torvalds

6

Linux

- A small Unix-like OS called MINUX was created by Andrew Tanenbaum for education purpose
- Triggered by the MINUX, Linus Torvalds, a Finnish graduate student, started a personal project, the Linux operating system in 1991
- The Kernel version 1.0 was released in 1994 and today the most recent stable version is 5.x (as of 2020)
- Developed under the GNU General Public License , the source code for Linux is freely available to everyone
- Today, Linux is maintained by Linus Torvalds with contributions from thousands of developers around the world
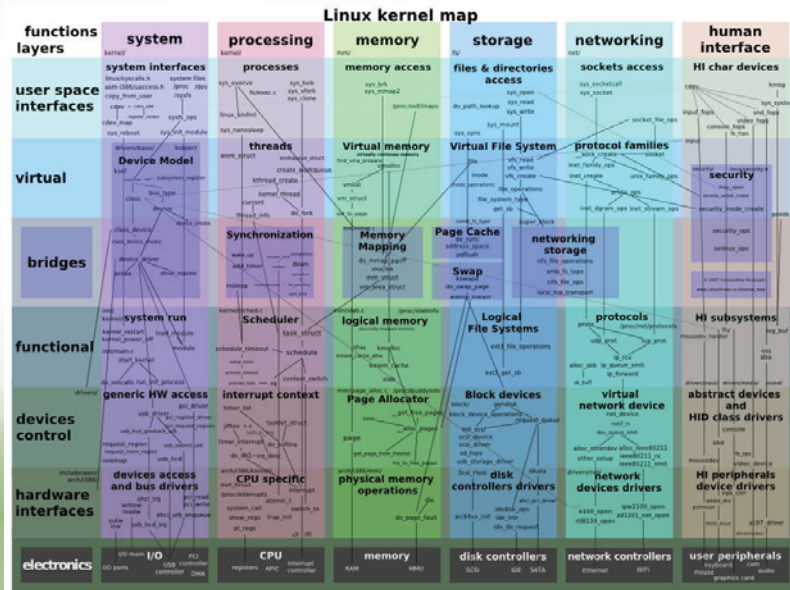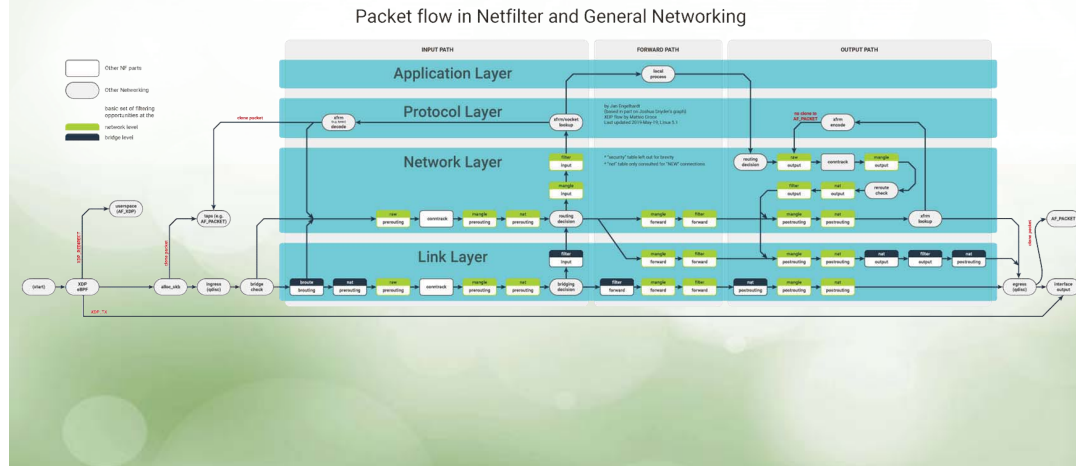
7

Linux Distributions
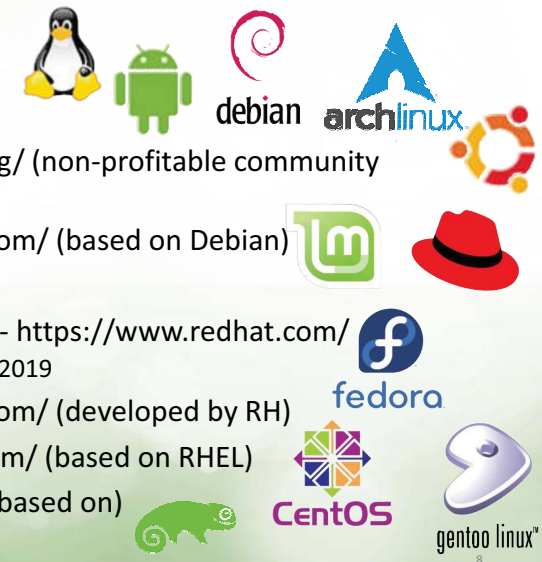
- Debian - https://www.debian.org/ (non-profitable community distribution)
- Ubuntu - https://www.ubuntu.com/ (based on Debian)
- Linux Mint - (based on Debian)
- Red Hat Enterprise Linux (RHEL) - https://www.redhat.com/
  - Red Hat was purchased by IBM in 2019
- Fedora - https://fedora.redhat.com/ (developed by RH)
- CentOS - https://www.centos.com/ (based on RHEL)
- SUSE - https://www.suse.com/ (based on)

8

Linux kernel

Example: Networking - Packet Loss

Packet flow in Netfilter and General Networking

Kernel Structure and User Levels

Least privileged

Most privileged

| Run Level	Mode	Action
0	Halt	Shuts down system
1	Single-User Mode	Does not configure network interfaces
2	Multi-User Mode	Does not configure network interfaces.
3	Multi-User Mode with Networking	Starts the system normally.
4	Undefined	Not used/User-definable
5	X11	As runlevel 3 + display manager(X)
6	Reboot	Reboots the system
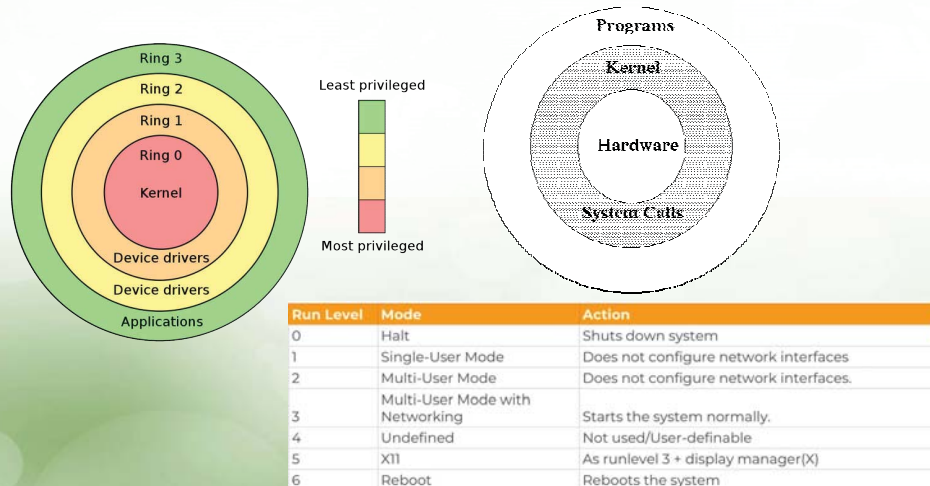
A Typical File System

- Usually draw a file system an inverted tree structure
- The **root directory**, denoted by /, is at the top and work down through sub-directories underneath it
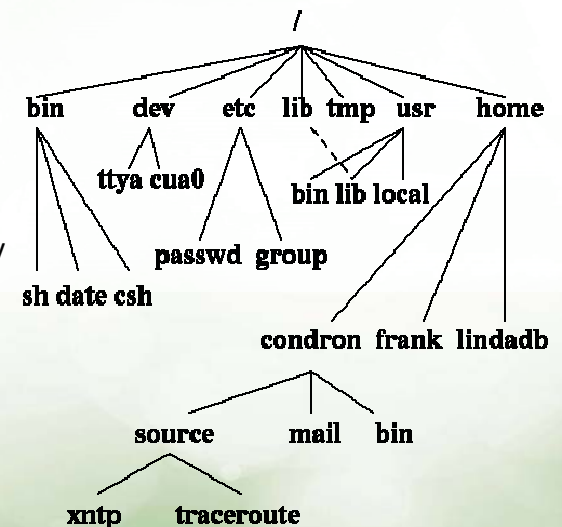
File System (cont'd)

- Each node is either a file or a directory of files
- A directory further can contain other files and directories
- Specify a file or directory by its path name, either the full, or absolute, path name or the one relative to a location
- The full path name must starts with the root, /, and follows the branches of the file system, each separated by /, until you reach the desired file, e.g.:
  - /home/condron/source/xntp

File System (cont'd)

- A relative path name specifies the path relative to another, usually the current working directory that you are at
- Two special directories :
  - `.` the current directory
  - `..` the parent of the current directory
- If currently at /home/frank, specify the path above in a relative fashion is
  - ../condron/source/xntp
- This indicates that firstly goes up one directory level, then goes down through the condron directory, then the source directory and then to xntp

File System (cont'd)

- / The root of all directories on the system; all other directories are subdirectories of this directory, either directly or through other subdirectories
- /bin Essential tools and other programs (or binaries)
- /dev Files representing the system's various hardware devices, e.g., use the file "/dev/cdrom" to access the CD-ROM drive
- /etc Miscellaneous system configuration files, startup files, etc

File System (cont'd)

- /home The home directories for all of the system's users
- /lib Essential system library files used by tools in "/bin"
- /proc Files that give information about current system processes
- /root The super-user's home directory, whose username is root

(In the past, the home directory for the super-user was simply "/"; later, "/root" was adopted for this purpose to reduce clutter in "/")

File System (cont'd)

- `/sbin` Essential system administrator tools, or system binaries
- `/tmp` Temporary files
- `/usr` Subdirectories with files related to user tools and applications

Inodes

- Every directory and file is listed in its parent directory
- For the root directory, the parent is itself
- A directory is a file that contains a table listing the files contained within it, giving file names to the **inode** (index node) numbers in the list
- The information about all the files and directories is maintained in the *INODE TABLE*
- An *inode* is an entry in the table containing information about a file (metadata) including file permissions, UID, GID, size, time stamp, pointers to files data blocks on the disk etc

Users and Groups

- In Linux, there is a concept of user and an associated group
- System determines whether or not a user or group can access a file or program based on the permissions assigned to them
- Apart from all the users, there is a special user called Super User or the root (the user) which has permission to access any file and directory

Access Permissions

- There are three permissions for any file, directory or application program
- The following lists the symbols used to denote each, along with a brief description:
  - *r*: Indicates that a given category of user can read a file
  - *w:* Indicates that a given category of user can write to a file
  - *x:* Indicates that a given category of user can execute the file

File Ownerships and Permissions

- Each of the three permissions are assigned to three defined categories of users
- The categories are:
  - owner: The owner of the file or application
  - group: The group that owns the file or application
  - others: All users with access to the system

Showing File Ownerships

- `$ ls` is used to list the contents of a directory
- If the command `ls` is written with parameter `-l` then the command lists contents of the working directory with details

  ```
  $ ls –l
  ```

List Files

- One can easily view the permissions for a file by invoking a long format listing using the command

  ```
  $ ls -l
  ```

- For instance, if the user juan creates an executable file named `test`, the output of the command `$ ls -l test` would look like this

  ```
  -rwxrwxr-x 1 juan student 0 Sep 26 12:25 test
  ```

File Permissions

- Permissions for this file are listed are listed at the start of the line, starting with **rwx**
- This first set of symbols define owner access
- The next set of **rwx** symbols define group access
- The last set of symbols defining access permitted for all other users

File Permissions (cont'd)

- This listing indicates that the file is readable, writable, and executable by the user who owns the file (user juan) as well as the group owning the file (which is a group named student)
- The file is also world-readable and world-executable, but not world-writable

Some Commands

- `$ cd try_it`
   - Changes the directory to try_it
- `$ pwd`
   - Prints present working directory (e.g. /home/smith/try_it)
- `$ cd ..`
   - Move up one directory
- `$ pwd`
   - Prints /home/smith

Some Commands (cont'd)

- *$ cd /home*
   - Set the absolute path
- pwd : Prints /home
- *$ cd*
   - Returned to the user's home directory
- $ pwd : Print /home/smith

Some Commands (cont'd)

- `$ mkdir my_dir`
   - Makes a new directory named "my_dir" (the path is given relative) as a subdirectory of the current directory
- `$ rmdir your_dir`
   - Removes directory your_dir if it is empty

Some Commands (cont'd)

- `$ cp file_1 file_2`
  - Copies file_1 to file_2. The both files must be in the same working directory. If they are in various directories, the path must be given.
- `$ mv file_1 file_2`
  - Moves file_1 to file_2, files must be in the same working directory.
  - For setting in different directories, the path(s) must be given
  - The file_1 is removed from the disk.
  - This command is also suitable for moving directories.

Removing Files

- `$ rm file_a`
  - Removes the file_a from the system
  - Can use *wildcard*, e.g.,
- `$ rm h*c`
  - Remove all files beginning with h and ending with c which are in working directory.
  - If writing
- `$ rm *`
  - Erases all files from your working directory.

Changing Ownership and Group

- `$ chown <owner> <file/directory name>`
  - Ownership of a file or directory can be changed with command
- `$ chgrp <group> <file/directory name>`
  - Group of a file or directory can be changed with command
- `$ chmod -R ### <filename or directory>`
  - Permissions of a file can be changed with the `chmod` command
  - `-R` is optional and when used with directories will traverse all the sub-directories of the target directory changing ALL the permissions to ###

Setting RWX

- The #'s can be:
- 0 = Nothing
  1 = Execute
  2 = Write
  3 = Execute & Write? (2 + 1)
  4 = Read
  5 = Execute & Read (4 + 1)
  6 = Read & Write (4 + 2)
  7 = Execute & Read & Write (4 + 2 + 1)

Practice

- Login
- Find the present Directory
- Write the root directory structure
- Write a few commands available in /bin and /sbin directory
- Find your directory
- Write the permissions of your home directory
- Create a new Directory test in your directory
- Copy the file /etc/resolv.conf in your directory
- Rename the test directory to testing
- Delete the testing directory
- Change the permissions of your directory to 700
- Change the permissions of /tmp directory to 700

Thanks