

# UNIX/LINUX OPERATING SYSTEM

By

Subrahmanya Bhat

Dept. MCA, Srinivas University

- ▣ Introduction to Unix
- ▣ History of UNIX
- ▣ What is LINUX
- ▣ LINUX Distributions
- ▣ Unix OS Structure
- ▣ Unix File System
- ▣ Unix Directories, Files and Inodes
- ▣ Users, Groups and Permissions

# UNIX

- ▣ Unix is a multi-user, multi-tasking operating system.
- ▣ You can have many users logged into a system simultaneously, each running many programs.
- ▣ It's the kernel's job to keep each process and user separate and to regulate access to system hardware, including cpu, memory, disk and other I/O devices.

# History of UNIX

- ▣ First Version was created in Bell Labs in 1969.

# History of UNIX

Some of the Bell Labs programmers who had 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. It was given the name UNIX by Brian Kernighan.

# History of UNIX

- ▣ 1973 Unix is re-written mostly in C, a new language developed by Dennis Ritchie.
- ▣ Being written in this high-level language greatly decreased the effort needed to port it to new machines.

# History of UNIX

- ▣ 1977 There were about 500 Unix sites world-wide.
- ▣ 1980 BSD 4.1 (Berkeley Software Development)
- ▣ 1983 SunOS, BSD 4.2, System V
- ▣ 1988 AT&T and Sun Microsystems jointly develop System V Release 4 (SVR4). This later developed into UnixWare and Solaris 2.
- ▣ 1991 Linux was originated.

# What is LINUX

- ▣ Linux is a free Unix-type operating system originally created by Linus Torvalds with the assistance of developers around the world.
- ▣ It originated in 1991 as a personal project of Linus Torvalds, a Finnish graduate student.



# What is LINUX..

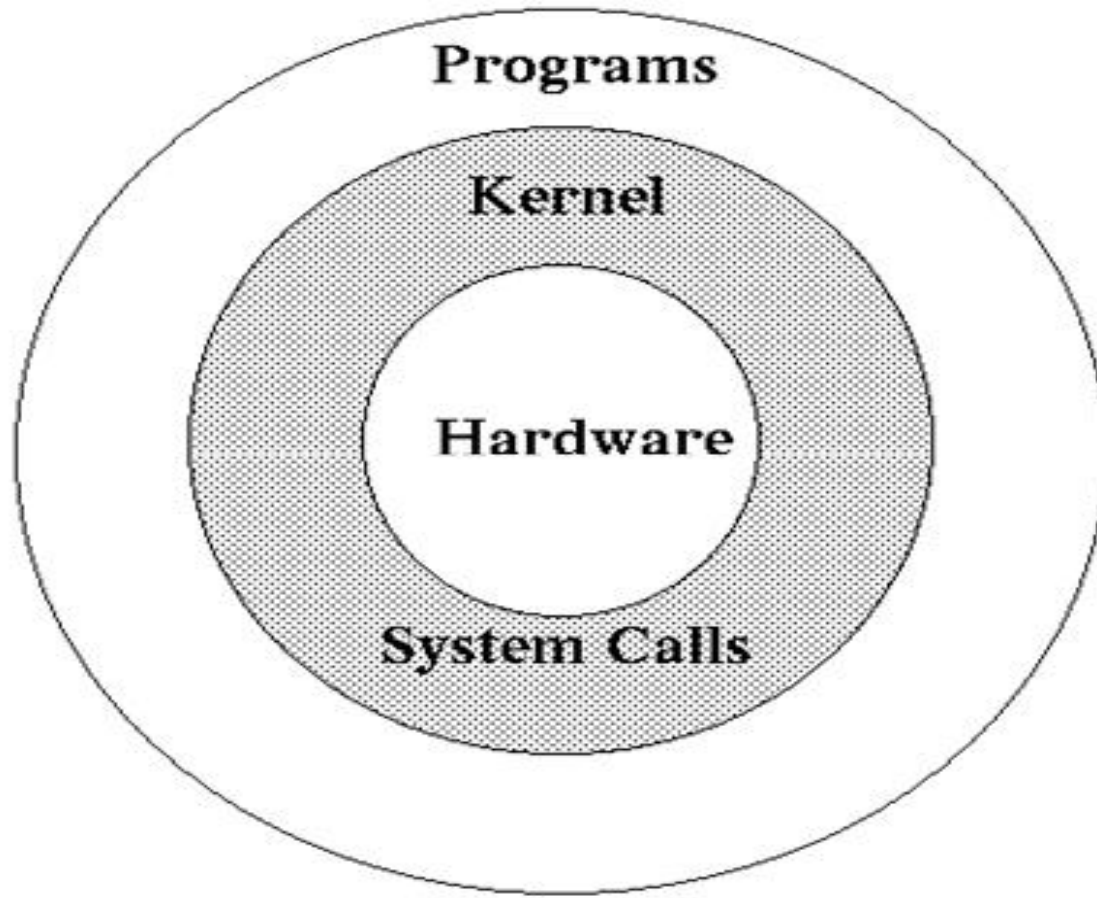
- ▣ The Kernel version 1.0 was released in 1994 and today the most recent stable version is 2.6.9
- ▣ Developed under the GNU General Public License, the source code for Linux is freely available to everyone.

# LINUX Distributions

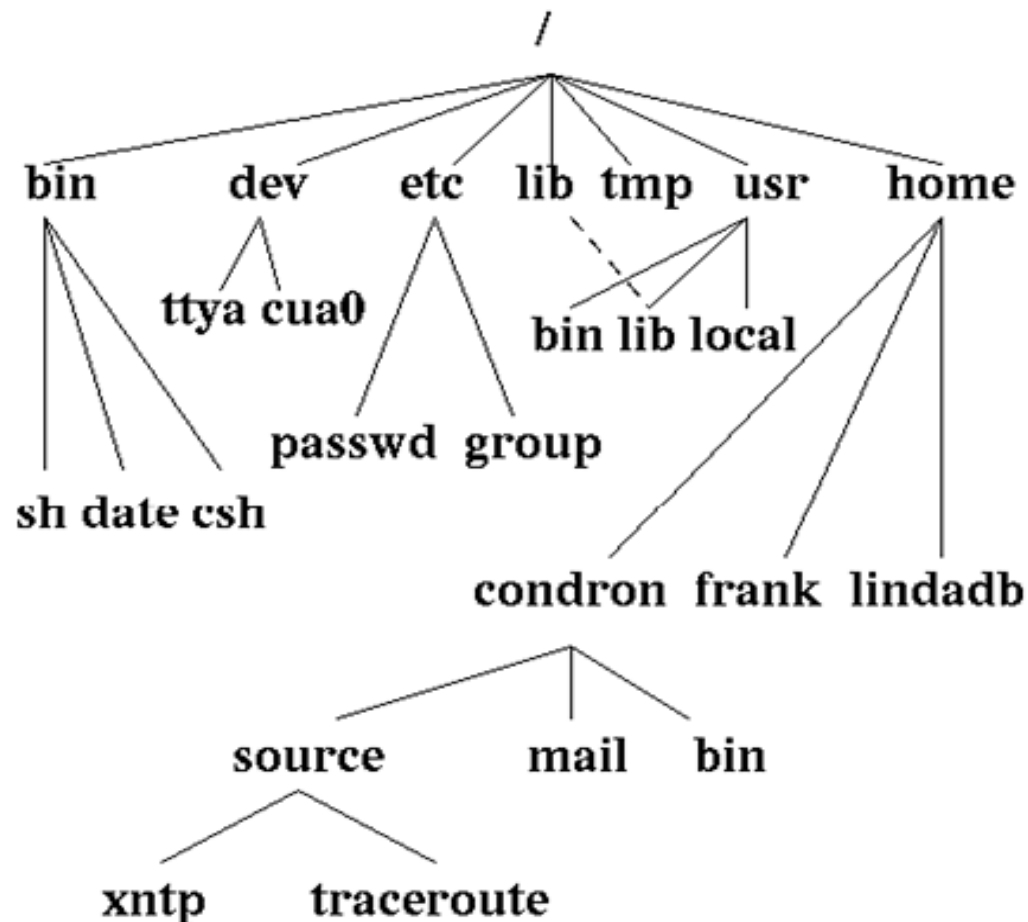
- ▣ Mandrake: <http://www.mandrakesoft.com/>
- ▣ RedHat: <http://www.redhat.com/>
- ▣ Fedora: <http://fedora.redhat.com/>
- ▣ SuSE/Novell: <http://www.suse.com/>
- ▣ Debian: <http://www.debian.org/>

- ▣ Red Hat Enterprise Linux is a Enterprise targeted Operating System. It based on mature Open Source technology and available at a cost with one year Red Hat Network subscription for upgrade and support contract.

# UNIX Structure



# UNIX File System



# File System

- ▣ The Unix file system looks like an inverted tree structure.
- ▣ You start with the root directory, denoted by `/`, at the top and work down through sub-directories underneath it.

# File System..

- ▣ Each node is either a file or a directory of files, where the latter can contain other files and directories.
- ▣ You specify a file or directory by its path name, either the full, or absolute, path name or the one relative to a location.

# File System..

- ▣ The full path name 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

- ▣ 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

- ▣ So if I'm at `/home/frank` and wish to specify the path above in a relative fashion I could use:

`../condron/source/xntp`

- ▣ This indicates that I should first go up one directory level, then come down through the condron directory, followed by the source directory and then to xntp.

# Structure of Standard Directories in Unix/Linux

- ▣ / The ancestor 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).

# Structure of Standard Directories in Unix/Linux..

- ▣ /dev Files representing the system's various hardware devices. For example, you use the file ``/dev/cdrom'` to access the CD-ROM drive.
- ▣ /etc Miscellaneous system configuration files, startup files, etc.

# Structure of Standard Directories in Unix/Linux..

- ▣ /home The home directories for all of the system's users.
- ▣ /lib Essential system library files used by tools in `/bin'.

# Structure of Standard Directories in Unix/Linux..

- ▣ /proc Files that give information about current system processes.
- ▣ /root The superuser's home directory, whose username is root. (In the past, the home directory for the superuser was simply '/'; later, '/root' was adopted for this purpose to reduce clutter in '/'.)

# Structure of Standard Directories in Unix/Linux..

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



# Directories, Files

- ▣ Every directory and file is listed in its parent directory.
- ▣ In the case of the root directory, that parent is itself.
- ▣ A directory is a file that contains a table listing the files contained within it, giving file names to the inode numbers in the list.

# Inodes

- ▣ The information about all the files and directories is maintained in INODE TABLE
- ▣ An Inode (Index Nodes) 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, Groups and Access Permissions

- ▣ In UNIX/LINUX, there is a concept of user and an associated group
- ▣ The 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 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, w & x

# Access Permissions..

- ▣ 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.

# Access Permissions..

- ▣ Each of the three permissions are assigned to three defined categories of users.
  
- ▣ The categories are:
  - Owner
  - Group
  - Others

# Access Permissions..

- ▣ 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.



# Access Permissions..

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

- ▣ The permissions for this file are listed at the start of the line, starting with `rw`.
- ▣ This first set of symbols define owner access.
- ▣ The next set of `rw` symbols define group access
- ▣ The last set of symbols defining access permitted for all other users.

# Listing the Content of a Directory

- ▣ 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**

# Moving in Directories

- ▣ `cd try_it`

Changes the directory to try\_it

- ▣ `Pwd`

Prints present working directory (e.g. /home/smith/try\_it)

- ▣ `cd ..` Move to superior directory
- ▣ `pwd` /home/smith
- ▣ `cd /home` The absolute path
- ▣ `Pwd` /home
- ▣ `cd` The system is returned to the user home directory
- ▣ `pwd` /home/smith

# Make Directory

- ▣ `mkdir my_dir`

makes new directory `my_dir` as a subdirectory of the current directory.

- ▣ `rmdir your_dir`

removes directory `your_dir` , if it is empty

- ▣ `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. The both files must be in the same working directory.

The file\_1 is removed from the disk.



- ▣ `rm file_a`

removes the `file_a` from the system

- ▣ `rm h*c`

- ▣ `rm *`

# Access Permission of File/Directory

- ▣ The ownership of the file or directory can be changed using the command

**chown <owner> <file/directory name>**

# Access Permission of File/Directory..

- ▣ The group of the file or directory can be changed using the command

`chgrp <group> <file/ directory name>`

# Access Permission of File/Directory..

- ▣ The permissions of the file can be changed using chmod command

**chmod -R ### <filename or directory>**

-R is optional and when used with directories will traverse all the sub-directories of the target directory

# Access Permission of File/Directory

▣ 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)

# Assignment 1 Mark

- ▣ 1) What is a OS?
- ▣ 2) Why the name Linux has derived?
- ▣ 3) What is a notation used for current directory in Linux?
- ▣ 4) What is a notation used for Parent directory in Linux?
- ▣ 5) When was the first Linux has been released?
- ▣ 6) What is a significance of /bin directory?
- ▣ 7) What is the difference between cp and mv command?

# Assignment 7 Mark

- ▣ 1) What is Linux? Give the characteristics of OS?
- ▣ 2) Describe the file system structure in Linux.
- ▣ 3) What is the difference between naming the file with absolute path and relative path? Illustrate with an example.
- ▣ 4) Give the different levels of security implementation available with Linux O.S.
- ▣ 5) What is the usage of chown and chmod commands in Linux?
- ▣ 6) List the different operations and their equivalent decimal values on a file for different users.
- ▣ 7) Give the features of Linux O.S.