Introduction to Linux

Prachi Pandey
C-DAC Bangalore

Introduction

- Developed by Linus Torvalds
- Initial release in1991
- Used in most of the computers, ranging from super computers to embedded system
- Multi user
- Multi tasking
- Monolithic kernel

FOSS

- Free Open Source Software
- Free – No charge for using it
- Open source – Source code is available and any body can contribute to the development. Organization independent

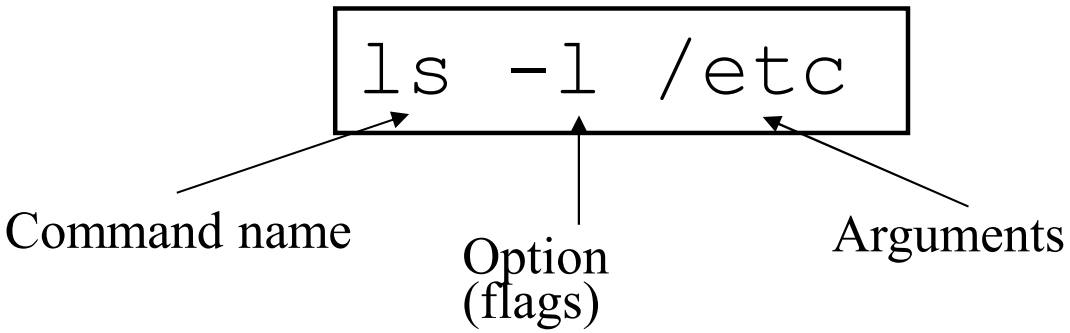
Why Linux

- Growing popularity
- Powerful
  - Runs on multiple hardware platforms
  - Users like its speed and stability
  - No requirement for latest hardware
  - Secure
- Its free
  - Vendors are distributors who package Linux

Linux Distributions

Getting started with Linux

- Logging in
- Commands
- Command structure



Linux File System

- Everything is a file
  - General files
  - Directory files
  - Device files
- No drives
- Peripherals like hard drives, cd rom, printers are also considered files

File Structure

Tree like hierarchical file system

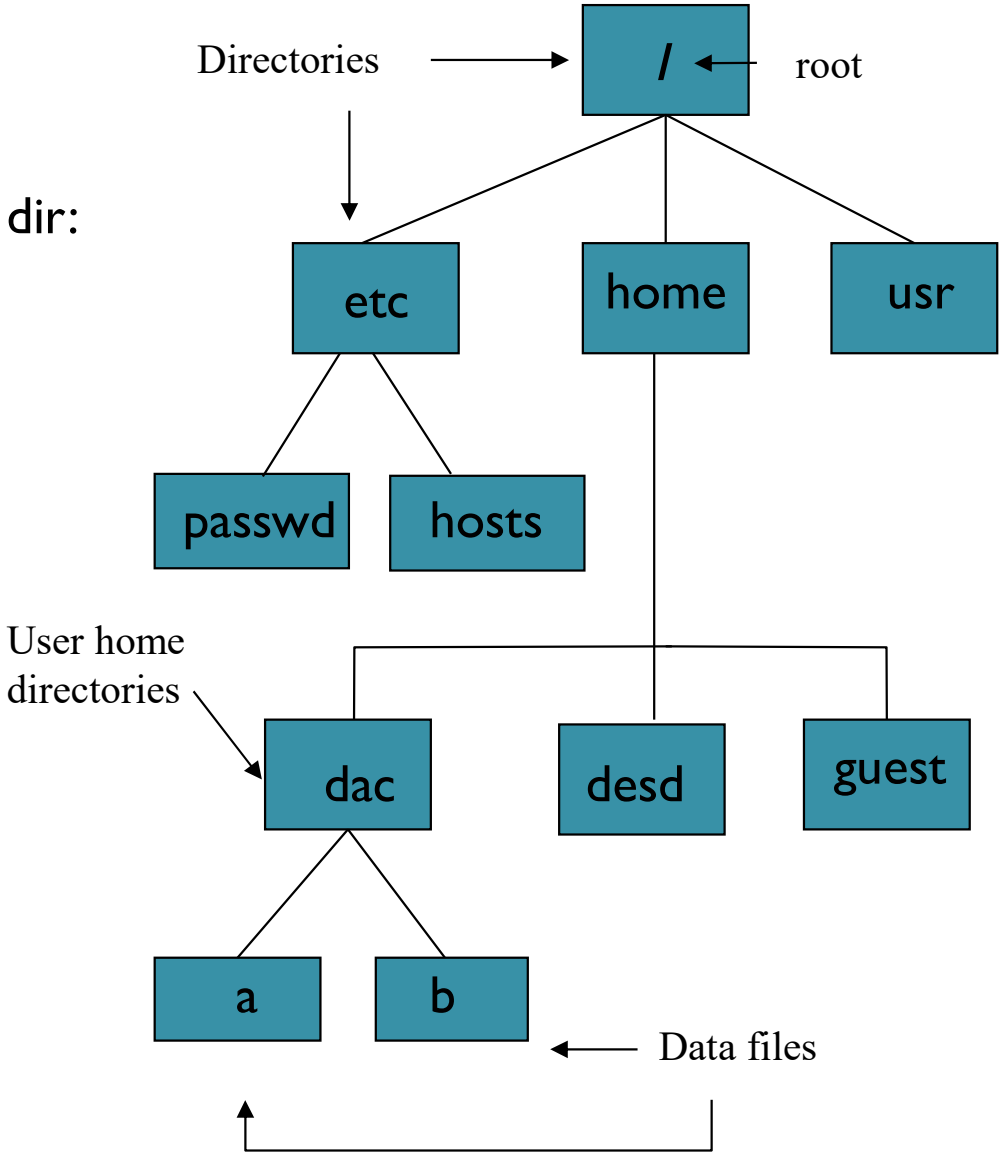
Root is the main directory (denoted with /)

Special files

- **/home**
- **/bin, /usr/bin**
- **/sbin, /usr/sbin**
- **/etc**
- **/lib, /usr/lib**
- **/var**
- **/dev**
- **/tmp**
- **/opt**

File Structure

- Absolute path
  - To access file a from current dir:
    /home/dac/a
- Relative path
  - To access file dac from a:
    ../dac



Linux FS vs Windows FS

Linux vs. Windows File Structure

- In Linux, there are no drives like C:, D:
- Linux files are ordered in a tree structure
- Top hierarchy is /
- Path separator is / not \
- File extensions do not have any meaning

User accounts

- 3 types of user accounts
  - Regular
  - Root
  - Service account
- Root user is super user and has all admin privileges
- For every user, /home/<username> directory is created which is called home directory

Common Commands

- **pwd**
- **cat**
- **echo**
- **man**
- **cd <*dir*>**
- **ls**

Common Commands

Command	Description
cd or cd ~	Navigate to HOME directory
cd ..	Move one level up
cd	To change to a particular directory
cd /	Move to the root directory

ls -l

1st Column	File type and access permissions
2nd Column	# of HardLinks to the File
3rd Column	Owner and the creator of the file
4th Column	Group of the owner
5th Column	File size in Bytes
6th Column	Date and Time
7th Column	Directory or File name

File commands

- **cp <*fromfile*> <*tofile*>**
- **mv <*fromfile*> <*tofile*>**
- **rm <*file*>**
- **mkdir <*newdir*>**
- **rmdir <*dir*>**

Authorization in Linux

- Linux divides authorization in two levels
  - Ownership
  - Permission

- There are 3 user types on a Linux system
  - Owner (user)
  - Group
  - Others

File Permissions

- Every file
  - Is owned by someone
  - Belongs to a group
  - Has certain access permissions for owner, group, and others
- Every user:
  - Has a uid (login name), gid (login group) and membership of a "groups" list:

File Permissions

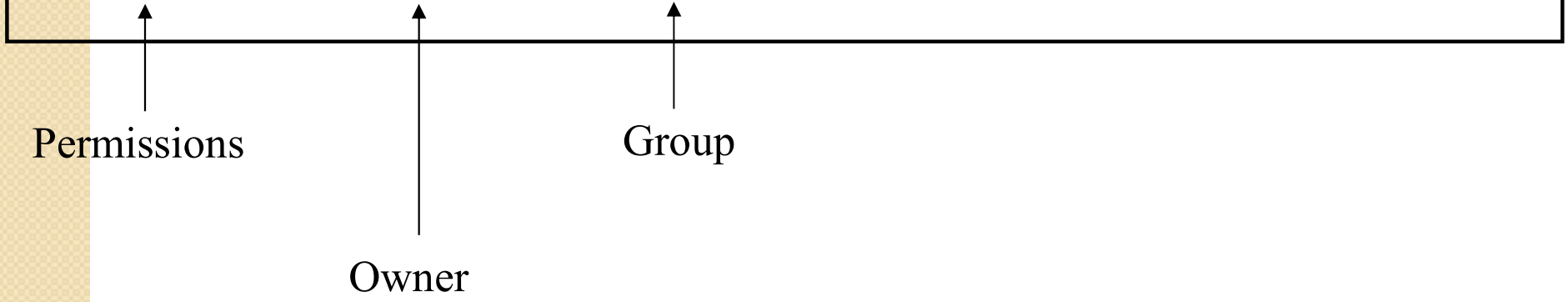
Linux provides three kinds of permissions:

- **Read** - users with read permission may read the file or list the directory (r)
- **Write** - users with write permission may write to the file or new files to the directory (w)
- **Execute** - users with execute permission may execute the file or lookup a specific file within a directory (x)

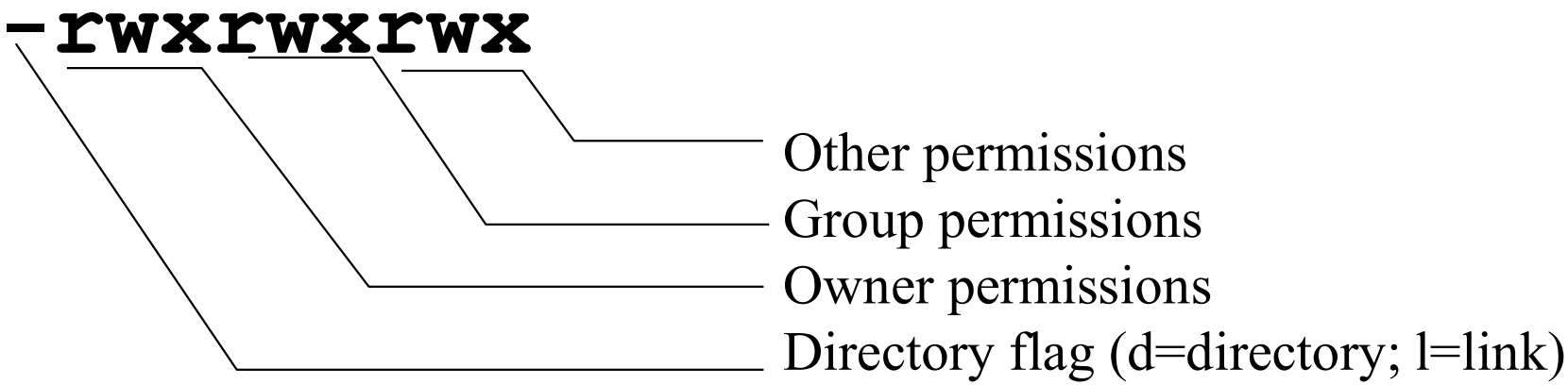
File Permissions

- The long version of a file listing (`ls -l`) will display the file permissions:

```
-rwxrwxr-x   1 rvdheij  rvdheij      5224 Dec 30 03:22 hello
-rw-rw-r--   1 rvdheij  rvdheij       221 Dec 30 03:59 hello.c
-rw-rw-r--   1 rvdheij  rvdheij      1514 Dec 30 03:59 hello.s
drwxrwxr-x   7 rvdheij  rvdheij      1024 Dec 31 14:52 posixuft
```



Interpreting File Permissions



Changing Permissions

- chmod
- chmod permissions filename
- Absolute and symbolic mode

Absolute(Numeric) mode

Number	Permission Type	Symbol
0	No Permission	---
1	Execute	--x
2	Write	-w-
3	Execute + Write	-wx
4	Read	r--
5	Read + Execute	r-x
6	Read +Write	rw-
7	Read + Write + Execute	rwx

Symbolic mode

- To change the permissions for a specific owner

Operator	Description
+	Adds a permission to a file or directory
-	Removes the permission
=	Sets the permission and overrides the permissions set earlier.

User Denotations	
u	user/owner
g	group
o	other
a	all

Changing Ownership

- chown
- chown user <filename>
- chown user:group <filename>

- Groups
  - id
  - /etc/groups
  - chgrp group filename

Summary

- Linux being a multi-user system uses permissions and ownership for security.
- There are three user types on a Linux system viz. User, Group and Other
- Linux divides the file permissions into read, write and execute denoted by r,w, and x
- The permissions on a file can be changed by 'chmod' command which can be further divided into Absolute and Symbolic mode
- The 'chown' command can change the ownership of a file/directory. Use the following commands:
  chown user file or chown user:group file
- The 'chgrp' command can change the group ownership **chrgrp group filename**

Pipes, grep

- Pipes
  - "|" denotes pipe
  - Help combine two or more commands
  - Output of one command serves as input for the next command

- grep
  - Used to find strings and values in a text document
  - Present the result in a format you want
  - grep <search_string>

- Less, pg and more commands are used for dividing a long file into readable bits

Options with grep

Option	Function
-v	Shows all the lines that do not match the searched string
-c	Displays only the count of matching lines
-n	Shows the matching line and its number
-i	Match both (upper and lower) case
-l	Shows just the name of the file with the string

Sort command

- Sorting the contents of a file alphabetically
- sort <filename>

Option	Function
-r	Reverses sorting
-n	Sorts numerically
-f	Case insensitive sorting

More commands

- **wc** command is used to count lines, words and characters, depending on the option used.
  - usage: wc [options] [file name]

- Options:

  -l : Number of lines

  -w: Number of words

  -c: Number of characters

More commands

- **diff**
  - diff filename1 filename2

- **find**
  - find path –name "filename"
  - find / –name "*.log"

Network commands

- For communication with other devices
  - telnet
    - Used to connect to a remote Linux computer and work on it
    - telnet <IP address or hostname>

  - ssh
    - Securely connect to a remote Linux computer
    - More secure than telnet
    - ssh username@<IP address or hostname>

FTP

- FTP is file transfer Protocol for data transfer among computers
- Logging in and establishing secure connection with a remote host
- Upload and download files
- Navigating through directories
- Browsing contents of the directories

FTP

Command	Function
dir	Display files in the current directory of remote computer
cd "dirname"	change directory to "dirname" on remote computer
put file	upload 'file' from local to remote computer
get file	Download 'file' from remote to local computer
quit	Logout

SFTP

- FTP is not secure
  - Data sent is in clear text and not encrypted
- SFTP is Secure File Transfer Protocol
  - Adds a layer of security
  - Data is encrypted
  - Authenticates user and server

More network commands

- finger
  - Display information about the system users.
  - finger <username>
- scp
  - SCP (secure copy) is a command-line utility that allows you to securely copy files and directories between two locations.
  - scp file.txt remote_username@IP:/remote/directory
- ping
  - Check connections with a hostname or IP address
  - ping <IP address or hostname>

System variables – PS1

- **PS1** is the primary prompt string
  - Environment variable which contains the value of the default prompt.
  - For most Linux systems, the defaults values have [\u@\h \W]\$
    - u is the username
    - h is the hostname
    - W is the current working directory
  - To see default value of PS1,
    - echo $PS1
  - To change the value of PS1,
    - export PS1="new value"

```
\a      an ASCII bell character (07)
\d      the date in "Weekday Month Date" format (e.g., "Tue May
        26")
\D{format}
        the format is passed to strftime(3) and the result is
        inserted into the prompt string; an empty format results
        in a locale-specific time representation. The braces are
        required
\e      an ASCII escape character (033)
\h      the hostname up to the first `.'
\H      the hostname
\j      the number of jobs currently managed by the shell
\l      the basename of the shell's terminal device name
\n      newline
\r      carriage return
\s      the name of the shell, the basename of $0 (the portion
        following the final slash)
\t      the current time in 24-hour HH:MM:SS format
\T      the current time in 12-hour HH:MM:SS format
\@      the current time in 12-hour am/pm format
\A      the current time in 24-hour HH:MM format
\u      the username of the current user
\v      the version of bash (e.g., 2.00)
\V      the release of bash, version + patch level (e.g., 2.00.0)
\w      the current working directory, with $HOME abbreviated
        with a tilde (uses the value of the PROMPT_DIRTRIM vari-
        able)
\W      the basename of the current working directory, with $HOME
        abbreviated with a tilde
\!      the history number of this command
\#      the command number of this command
\$      if the effective UID is 0, a #, otherwise a $
\nnn    the character corresponding to the octal number nnn
\\      a backslash
\[      begin a sequence of non-printing characters, which could
        be used to embed a terminal control sequence into the
        prompt
```

System variables – PS2

- If the command is too long to fit in one line, it can be broken down into multiple lines by giving "\" at the end of each line
- **PS2**: environment variable which contains the value the prompt used for a command continuation interpretation.
- The default interactive **PS2** value prompt for a multi-line command is "**>**"
- export PS2="new value"

Some more commands

uname	.. know your machine's characteristics
logout	.. logs off system
cut	.. slitting a file vertically
cat	.. display/create files
wc	.. count lines, words, characters
gzip	.. compressing a file
gunzip	.. uncompressing a file
more	.. views a file, pausing every screenful
less	.. similar to more, more powerful
file	.. show file type
tail	.. show the last few lines of a file
head	.. show the beginning of a file
w	.. shows who is logged on and what they're doing
finger	.. shows more information about a user
df	.. shows disk space available on the system
du	.. shows how much disk space is being used by folders
bc	.. a simple calculator
cal	.. display calendar
date	.. display system date