



redhat
L I N U X

Linux System Admin I



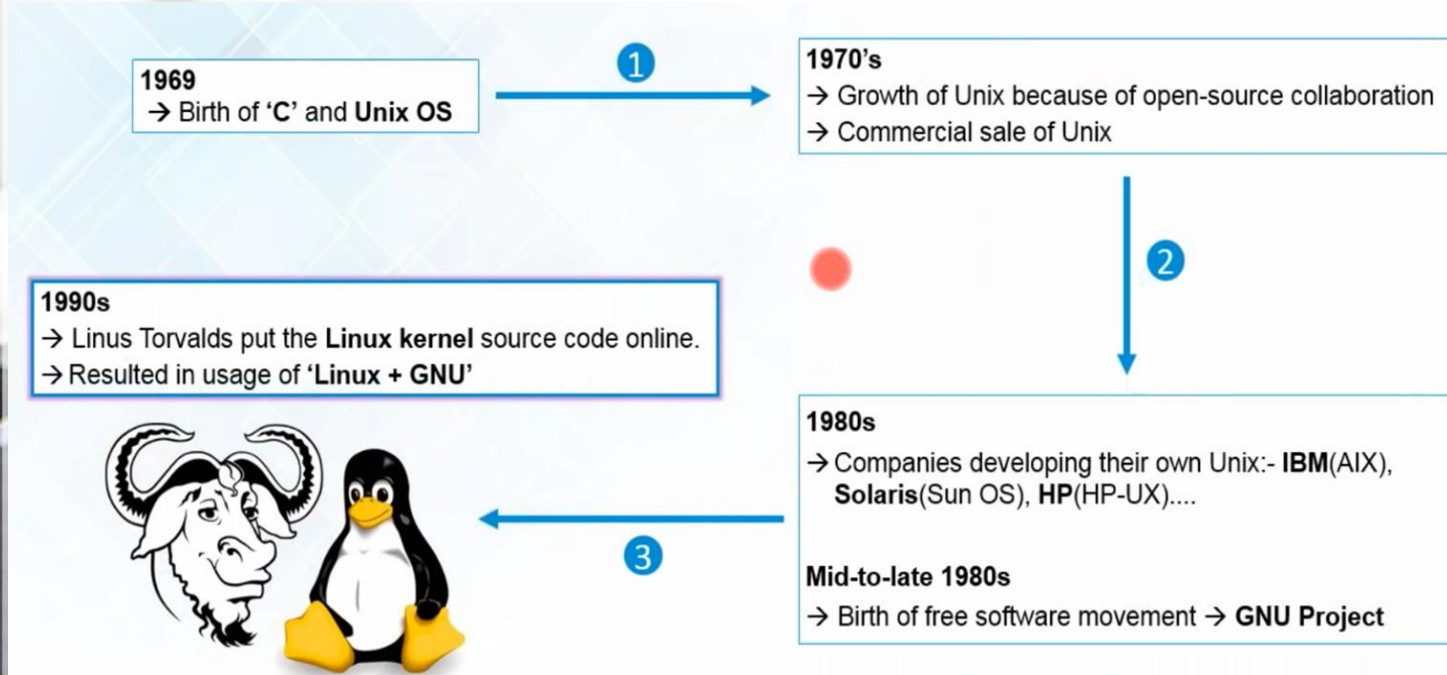
DAY 1 CONTENTS

- Linux History
- Linux Distribution
- What is FOSS
- Why Linux
- Linux Components
- Installation
- Accessing using command line
- Linux Documentation
- Managing files from The command line
- File and Directory Basics

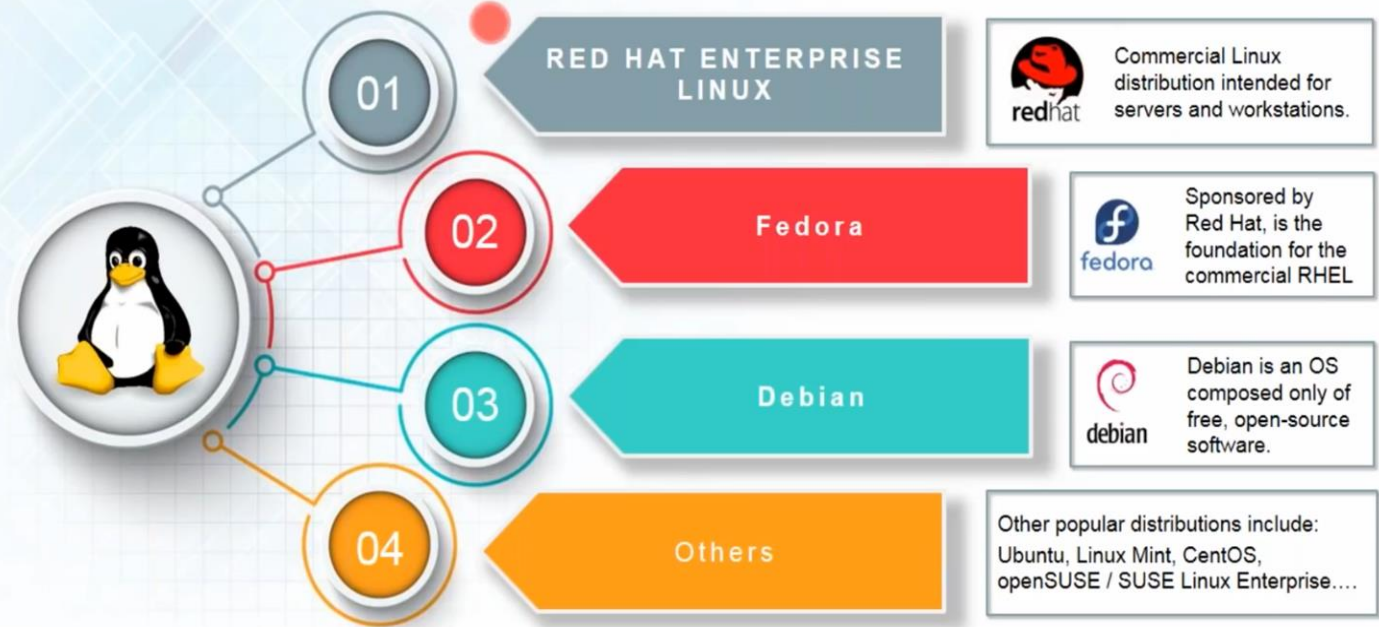
WHAT IS FOSS?

- Free/Open Source Software (FOSS) provides many freedoms, including the ability to:
 - View the source code used to compile programs
 - Make modifications
 - Distribute these modifications.
- Most FOSS is covered under a public license.

Linux History



Linux Distribution



Why Linux?



Linux Components

- **Kernel**

- Is the core of the operating system.
- Contains components like device drivers.
- It loads into RAM when the machine boots and stays resident in RAM until the machine powers off.

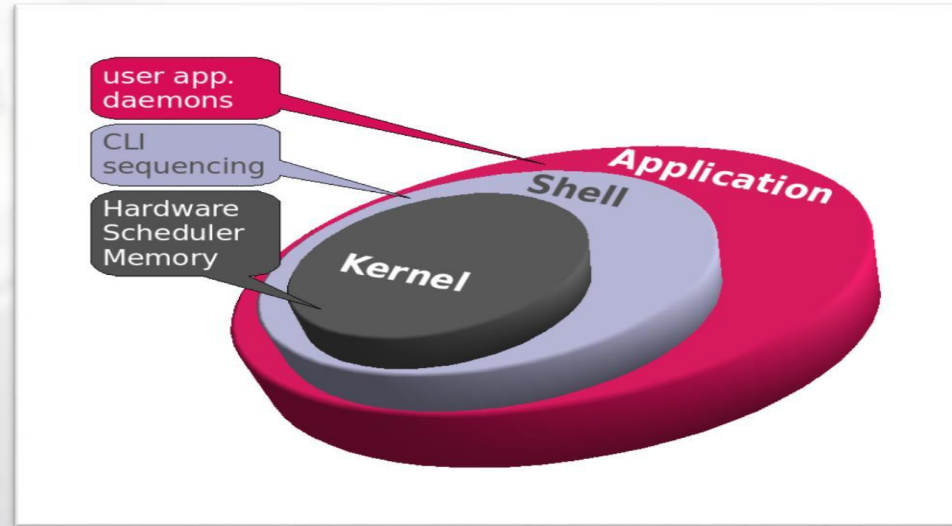
- **Shell**

- Provides an interface by which the user can communicate with the kernel.
- “bash” is the most commonly used shell on Linux.
- The shell parses commands entered by the user and translates them into logical segments to be executed by the kernel or other utilities.

- **Terminal**

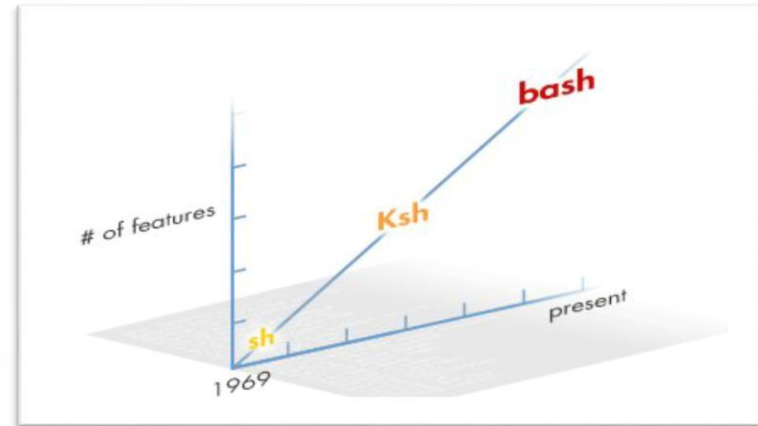
- Gives the shell a place to accept typed commands and to display their results

Linux Components



Command-Line Shell

- There are lot of shells as :
 - Bourn Shell (sh),
 - Korn Shell (ksh),
 - C Shell (csh) and
 - Bourn Again Shell (bash).



Running Commands

- Commands have the following syntax:
`command [options] [arguments]`
- Each item is separated by a space.
- Options modify the command's behavior.
- Arguments are files name or other information needed by the command.
- Separate commands with semicolon (;).

Examples

ls

To list the content of a directory

Ls -R

Will show you directories and subdirectories.

ls -a

Will list hidden file

Ls -aR

To list the content of directories and subdirectories with the hidden files

Examples

cal

September 2010

S	M	Tu	W	Th	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Examples

```
cal 5 2004
```

May 2004

S	M	Tu	W	Th	F	
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

```
cal; uname
```

```
cal 5 2002; date; uname
```

Interrupting Execution

- To interrupt a command that's taking too long to execute, use **[Ctrl]-c**.
- Occasionally, you might enter a command without an argument that expects input to come from the keyboard. In this case, use **[Ctrl]-d** to terminate the command.

Linux Documentation

Manual page consists of:

- Name
 - The name of the command and a one-line description
- Synopsis
 - The syntax of the command
- Description
 - Explanation how the command works and what it does
- Files
 - The file used by the command
- Bugs
 - Known bugs and errors
- See also
 - Other commands related to this one

Linux Documentation

man -k keyword

Shows the commands that have manual pages that contains any of the given keywords.

Linux Documentation

--help Option

- Another way to get help about a command.
- help is built in the command itself (if supported).

Directory

- Think of
 - File system as a building
 - Directory is a room
 - File is a desk
- The current working directory is the room you are.
- To find out where you are at any time

pwd

/home/guest



Changing Directory

- To move from directory to directory on the system

```
cd /home/user1/work cd ..
```

```
cd ~
```

```
cd -
```


Listing directory content

ls

```
dir1      dir2      file1
dir3      file2      file3
```

ls /home/user1/dir1

```
f1      f2
```

pwd

```
/home/user1
```

ls dir1

```
f1      f2
```

Listing directory content

```
ls -a dir1
```

```
.          .f1      f1
..         .f2      f2
```

```
ls -l dir1
```

```
total 2
```

```
--rw-r--r--    1  islam  islam  20  2 May  21  16:11  f1
--rw-r--r--    1  islam  islam  20  0 May  21  16:11  f2
```

```
ls -F
```

```
dir1/      dir2/      file1
dir3/      file2*      file3@
```

Listing directory content

```
ls -ld dir1
```

```
drwxr-xr-x 2 islam islam 20 512 May 21 16:06 dir1
```

```
ls -R
```

```
..:
```

```
dir1  dir3  dir2      file1
```

```
./dir1:  file2      file3
```

```
f1      f2
```

```
./dir2:
```

```
./dir3:
```

File Naming

- File names may be up to 255 characters.
- There are no extensions in Linux
- Avoid special characters as > < ? * # '
- File names are case sensitive

Viewing File Content

cat fname more

fname

- Scrolling keys for the more command
 - **Spacebar**: moves forward on screen
 - **Return**: scroll one line at a time
 - **b**: move back one screen
 - **/string**: search forward for pattern
 - **n**: find the next occurrence
 - **q**: quit and return to the shell prompt

head -n fname tail [-n|+n]

File Globbing

- When typing commands, it is often necessary to issue the same command on more than one file at a time.
- The use of wildcards, or “**metacharacters**”, allows one pattern to expand to multiple filenames

Meta Character

- Asterisk(*): represents 0 or more character, except leading (.)

Example:

ls f*

```
file.1 file.2 file.3 file4
```

```
file1 file2 file3 fruit
```

Meta Character

- Question mark(?) character represents any single character except the leading (.)

Examples

ls file?

file4 file1 file2

ls z?

z?: No such file or directory

- Square bracket([]): represent a range of characters for a single character position.

Example

ls [a-f]* ls

[pf]*

Meta Character

```
ls -a
```

```
. .. .profile abm bam bat battle project
```

```
ls -l b*
```

```
-rw-r----- 1 sgs 16 Feb 12 11:04 bam  
-rw-r----- 1 sgs 12 Feb 12 11:05 bat  
-rw-r----- 1 sgs 19 Feb 12 11:06 battle
```

```
ls *
```

```
abm bam bat battle project
```

```
ls .*
```

```
. .. .profile
```

```
ls *m
```

```
abm bam
```

Meta Character

ls **???** abm

bam bat **ls**

?a? bam

bat

ls **?a***

bam bat battle

ls ***a***

abm bam bat battle

Meta Character

```
ls [ab]*
```

```
abm bam bat battle
```

```
ls -l [ab]m
```

```
ls: "[ab]m: No such file or directory
```

```
ls [a-zA-Z]*
```

```
abm bam bat battle project
```

File & Dir Manipulating

- Copying Files and Directories

cp options source(s) target

Option	Description
-i	Prevents you from accidentally overwriting existing files or directories
-r	Copy a directory including the contents of all subdirectories

File & Dir Manipulating

- Moving and Renaming Files and Directories

mv options source(s) target

Option	Description
-i	Prevents you from accidentally overwriting existing files or directories

File & Dir Manipulating

- To create files

```
touch file(s)_name
```

- To create directories

```
mkdir [-p] dir(s)_name
```

File & Dir Manipulating

- To remove files

```
rm [-i] file(s)_name
```

- To remove directories `rmdir`

```
dir(s)_name  rm [-r]
```

```
dir(s)_name
```