



**RHSA1**  
**Red Hat System Administration I**  
**Day 1**

# Day 1 Contents

- Free/Open Source Software and Licenses.
- Linux History.
- Installation.
- Linux Components.
- Basic Commands.
- File and Directory Basics.
- Linux Documentation.



# 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. The most common public license is the GNU General Public License (GPL).**

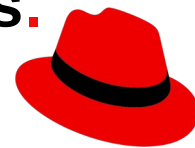
# FOSS Licenses

---

- An open-source license is a type of license for computer software and other products that allows the source code, blueprint or design to be used, modified and/or shared under defined terms and conditions.
- **Examples:**  
GPL, LGPL, Apache, Mozilla Public License and BSD.

# Linux History

- Multics was developed in the 1963-1969 period through the collaboration of the Massachusetts Institute of Technology (MIT), General Electric, and Bell Labs.
- Unix first version created in Bell Labs in 1969.
- Unix flavors:
  - ▶ IBM->AIX
  - ▶ Hewlett-Packard->HP/UX
  - ▶ Sun-> Solaris
  - ▶ Silicon Graphics->IRIX
- Operate in a same manner.
- Offer the same standard utilities and commands.



# Linux History

- In 1983, Richard Stallman started the GNU project with the goal of creating a **free UNIX-like** operating system.
  - ▶ GNU General Public License (**GPL**).
  - ▶ Free Software Foundation (**FSF**).
- In 1991, Linus Torvalds created Linux kernel.
- In 1992, **Linux and GNU developers** worked to integrate GNU components with Linux to make a fully functional and **free** operating system.

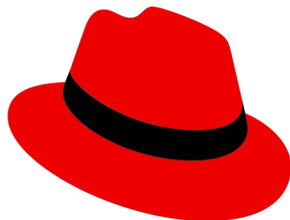
# Linux distributions



ubuntu



fedora



Red Hat



Solus



ZORIN



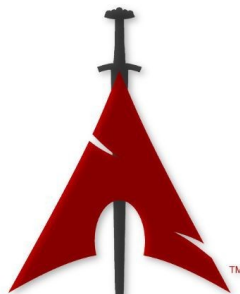
Linux Mint



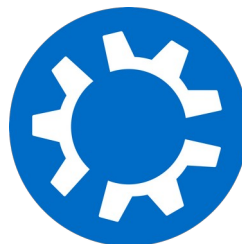
debian



archlinux



BlackArch



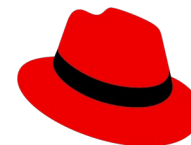
Kubuntu



CentOS



Manjaro



# Why Linux?

---

- Linux is **open source**.
- Linux is released under the **GNU General Public License (GPL)**.
- Linux is secure and virus free.
- Linux is perfect for programmers.
- Linux has a better community support.



# Why Linux?

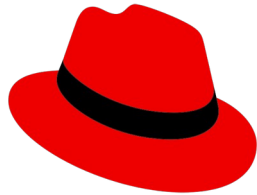
---

- **Linux is growing in the home users sector and the dominant of the professional and servers sector.**
- **Internet service providers (ISPs), e-commerce sites, and other commercial applications all use Linux today and continue to increase their commitment to Linux.**

# Red Hat

---

- Red Hat was founded on March 26, 1993.
- Red Hat Linux first appeared in 1994.



**Red Hat Enterprise  
Linux**



**fedora**

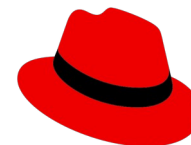
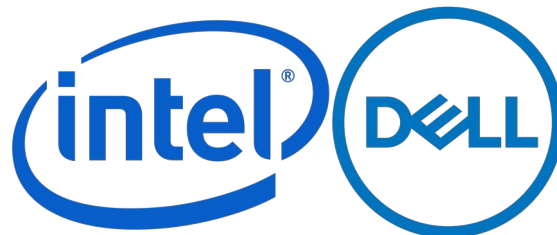
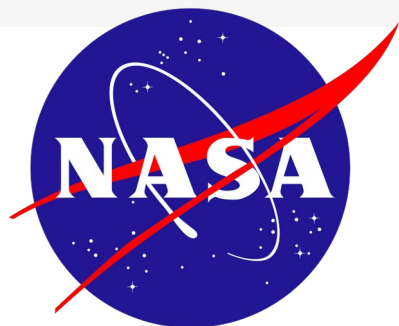


# Red Hat

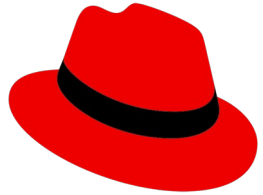
---

- **More than 90% of Fortune Global 500 companies use Red Hat products and solutions\*.**
- **The most demanding applications run better on Red Hat Enterprise Linux.**
- **RHEL scales well, and is more reliable.**
- **RHEL is secure.**
- **Red Hat partnership with hardware vendors.**
- **Red Hat training and support.**

# Red Hat



Red Hat



**Red Hat Enterprise  
Linux**



# Types of installation

---

- Automated installation.
- Graphical installation.
- Remote installation.

# 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.**

# Linux Components

- **Shell**

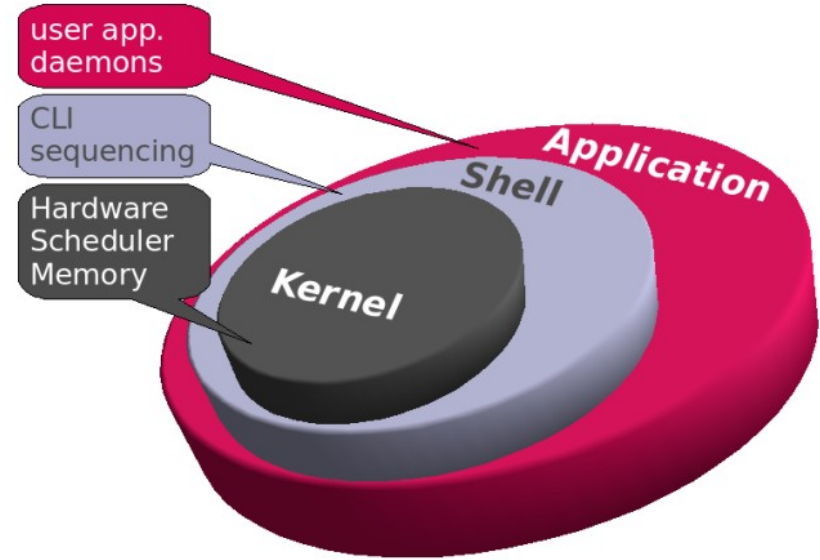
- ▶ Provides an interface by which the user can communicate with the kernel.
- ▶ “bash” is the most commonly used shell on Linux.
- ▶ The shell parses commands entered by the user and translates them into logical segments to be executed by the kernel or other utilities.



# Linux Components

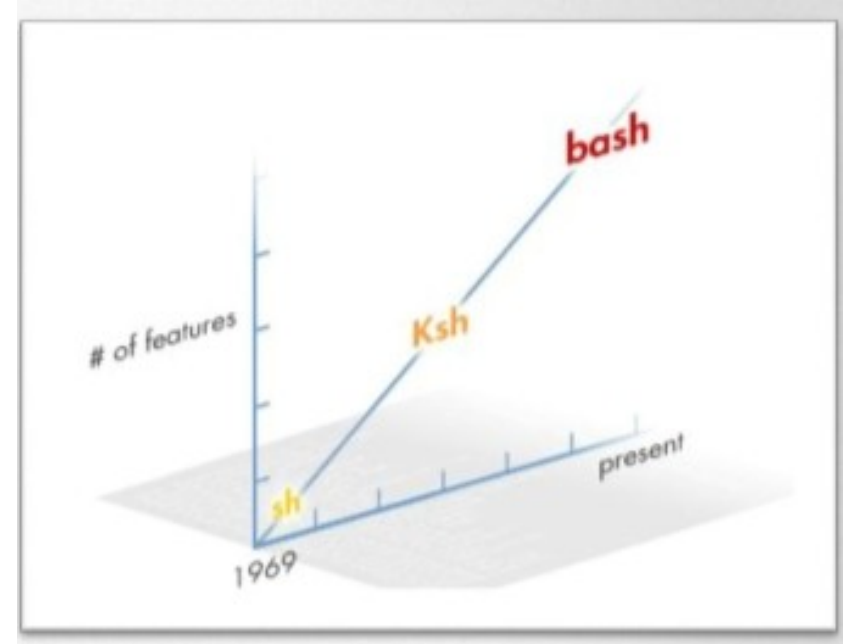
- **Terminal**

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



# Command-Line Shells

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



- They have different features that will be discussed later.

# 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

- `uname`  
`Linux`
- `uname -n`  
`host1`
- `uname -a`  
`Linux host1 .....`

[print system  
information]



# Examples

- **cal**

September 2020

**Su Mo Tu We Th Fr Sa**

		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			



# Examples

- **cal 9 2020**

September 2020

**Su Mo Tu We Th Fr Sa**

		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			

**[cal [month] [year]]**



# Examples

- **date**

**Tue Sep 1 19:26:42 EET 2020**

**print effective  
userid**

- **whoami**  
**root**

**print or set the  
system date and time**



**Red Hat**

# Directories

- 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`

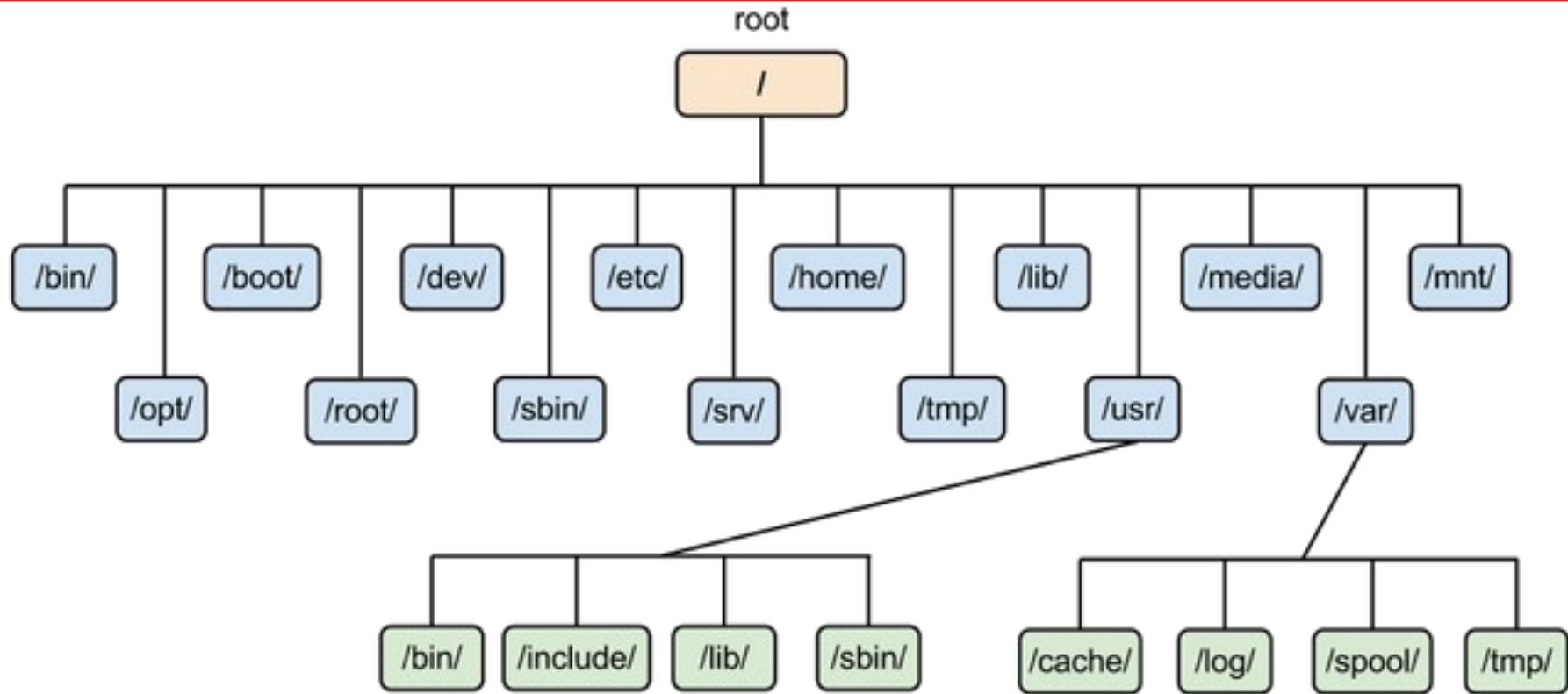
`/home/guest`

[print name of current/working directory]

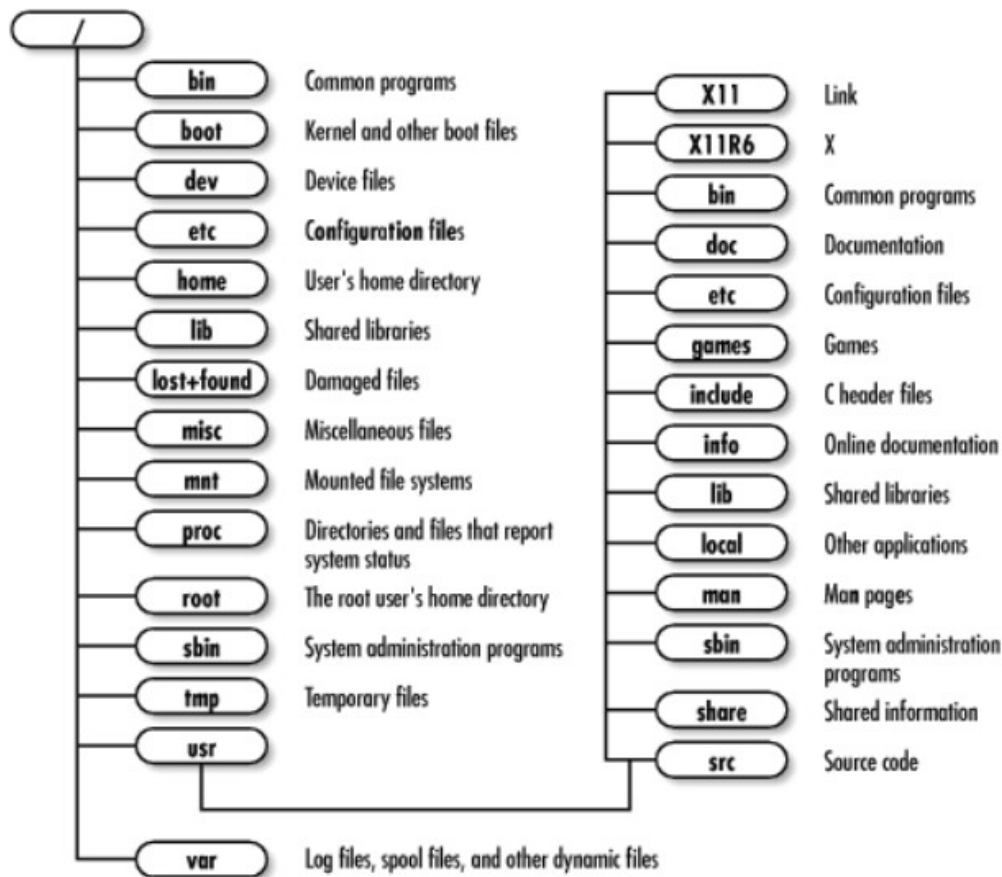




# File System



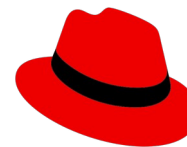
# Directories



- **Pathnames**

- ▶ **Absolute pathname**

- ▶ **Relative pathname**



**Red Hat**

# Changing Directories

- To move from directory to directory on the system.

`cd /home/user1/work`

`cd ..`

`cd ~`

`cd -`



# Listing Directory Contents

- **ls**

dir1 dir2 file1

dir3 file2 file3

- **ls /home/user1/dir1**

f1 f2

- **Pwd**

*/home/user1*

- **ls dir1**

f1 f2

[ **ls [option] [argument]** ]



# Listing Directory Contents

- **ls -a dir1**

. .f1 f1

.. .f2 f2

- **ls -l dir1**

total 2

-rw-r--r-- 1 fatma fatma 20 2 May 21 16:11 f1

-rw-r--r-- 1 fatma fatma 20 0 May 21 16:11 f2

- **ls -F**

dir1/ dir2/ file1

dir3/ file2\* file3@



# Listing Directory Contents

- **ls -ld dir1**

**d**rwxr-xr-x 2 fatma fatma 51237 May 29 16:06 dir1

**File  
type**

-:file  
d:directory  
c:?  
b:?  
S:?  
P:?  
l:?



**Red Hat**

# Listing Directory Contents

- **ls -R**

**.. :**

**dir1 dir2 file1**

**dir3 file2 file3**

**./dir1:**

**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 fname`

# File Globing

---

- 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.

# Metacharacters

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

**Example:**

**ls f\***

file.1 file.2 file.3 file4

file1 file2 file3 fruit

**ls \*3**

file.3 file3

dir3:

Moon planets space sun



# Metacharacters

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

Example:

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]\*



# Metacharacters

**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



# Metacharacters

Is ???

abm bam bat

Is ?a?

bam bat

Is ?a\*

bam bat battle

Is \*a\*

abm bam bat battle



# Metacharacters

**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. Manipulation

- Coping 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. Manipulation

- Coping Files and Directories

**cp** [options] sfile1 sfile2 sfile3 targetDest\_directory

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

# File & Dir. Manipulation

- 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. Manipulation

---

- To create files

**touch** file(s)\_name

- To create directories

**mkdir** [-p] dir(s)\_name

# File & Dir. Manipulation

---

- To remove files

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

- To remove directories

```
rmdir dir(s)_name
```

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

# 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.

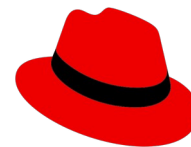
- **Bugs**

Known bugs and errors.

- **See also**

Other commands related to this one.

**[man man]**



**Red Hat**

# Linux Documentation

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

**man -k** keyword

**man -s** keyword

- Shows the commands one line description.

**whatis** command



# Linux Documentation

## --help Option

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



# 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.



# NTFS

---

- `sudo yum --enablerepo=extras install epel-release;`
- `sudo yum install ntfs-3g -y;`

# Grub

- Check that windows is detected. Run `grub2-mkconfig` but discard its output:
- `sudo grub2-mkconfig > /dev/null`  
Generating grub configuration file ...  
Found background image: `/usr/share/images/desktop-base/desktop-grub.png`  
Found linux image: `/boot/vmlinuz-3.16.0-4-amd64`  
Found initrd image: `/boot/initrd.img-3.16.0-4-amd64`  
Found memtest86+ image: `/boot/memtest86+.bin`  
Found memtest86+ multiboot image: `/boot/memtest86+_multiboot.bin`  
Found Windows 7 (loader) on `/dev/sda2`
- The output will look similar (but not identical) to what is shown above. Make sure that Windows is listed.

# Grub

- If Windows was listed in the previous step, go ahead and save the new configuration file. Make a backup first, just in case.
- `sudo cp /boot/grub2/grub.cfg /boot/grub2/grub.cfg.old`
- `sudo grub2-mkconfig -o /boot/grub2/grub.cfg`
- If all went well, you should now be able to reboot into Windows.

# Install Centos 8

---

- <https://www.linuxtechi.com/centos-8-installation-guide-screenshots/#:~:text=%2Fboot%20%E2%80%93%202%20GB%20>

# Linux File Hierarchy Structure

- <https://www.geeksforgeeks.org/linux-file-hierarchy-structure/>
- <https://www.howtogeek.com/117435/htg-explains-the-linux-directory-structure-explained/>