



# INTRODUCTION OF LINUX FILE SYSTEM

# LINUX

---

- The **Linux kernel** is an open-source monolithic unix-like computer operating system kernel.
- Its basic design from principles established in Unix during the 1970s and 1980s.
- Linux is a small UNIX system. Version 1.0 released in 1994, 2.2 in 1999.
- Today used on 7-10 million computers corporate computer network and Web servers .

# WHAT IS A FILE SYSTEM

---

- File System is responsible for storing information on disk and retrieving and updating this information.
- Example : FAT16, FAT32, NTFS  
ext2, ext3 ,ext4...
- In Linux everything is file.



# TYPES OF FILE SYSTEM

---

## 1.Network File System

- NFS
- SMB

## 2.Disk File System

- ext2
- ext3
- NTFS

# NETWORK FILE SYSTEM

---

Network File System are physically somewhere else, but appear as if they are mounted on one computer.

## ➤ NFS

It was developed by Sun.

## ➤ SMB

It was developed by Microsoft.

# PHYSICAL STRUCTURE ON THE DISK



- ❑ Boot Block :information needs to boot the system
- ❑ Super Block :File System Specifications Size Max. number of files  
Free blocks Free inodes
- ❑ inode List
- ❑ Block List :The files data



# IN OPERATION, LINUX KERNEL NEED TO KNOW

Where those data are store.

How it can be access.

And where we save the new data.

# SECOND STANDARD FILE SYSTEM (EXT2) 1993

- Maximum file size : 2TB
- Maximum volume size: 8 TB
- File name size: 255 characters
- Supports:
  - 1.POSIX Permissions
  - 2.File compression



# THIRD STANDARD FILE SYSTEM (EXT3)

---

- Maximum file size : 2 TB
- Maximum volume size: 2 TB
- File name size: 255 characters
- Supports:
  - 1.POSIX Permissions
  - 2.File compression

# RISAR FILE SYSTEM

---

➤ User Journaling

➤ Maximum file size : 8 TB

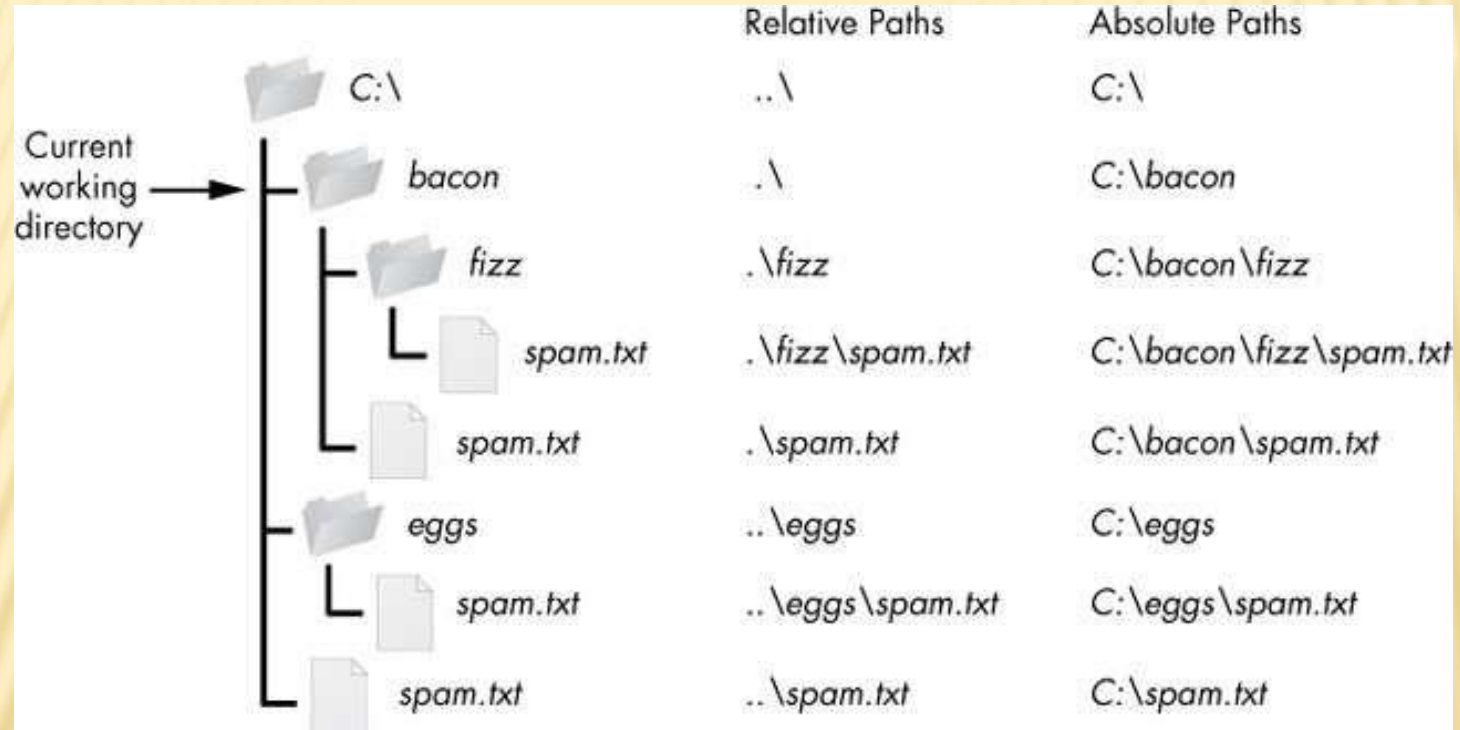
Maximum volume size: 16 TB

# FOURTH STANDARD FILE SYSTEM (EXT4)

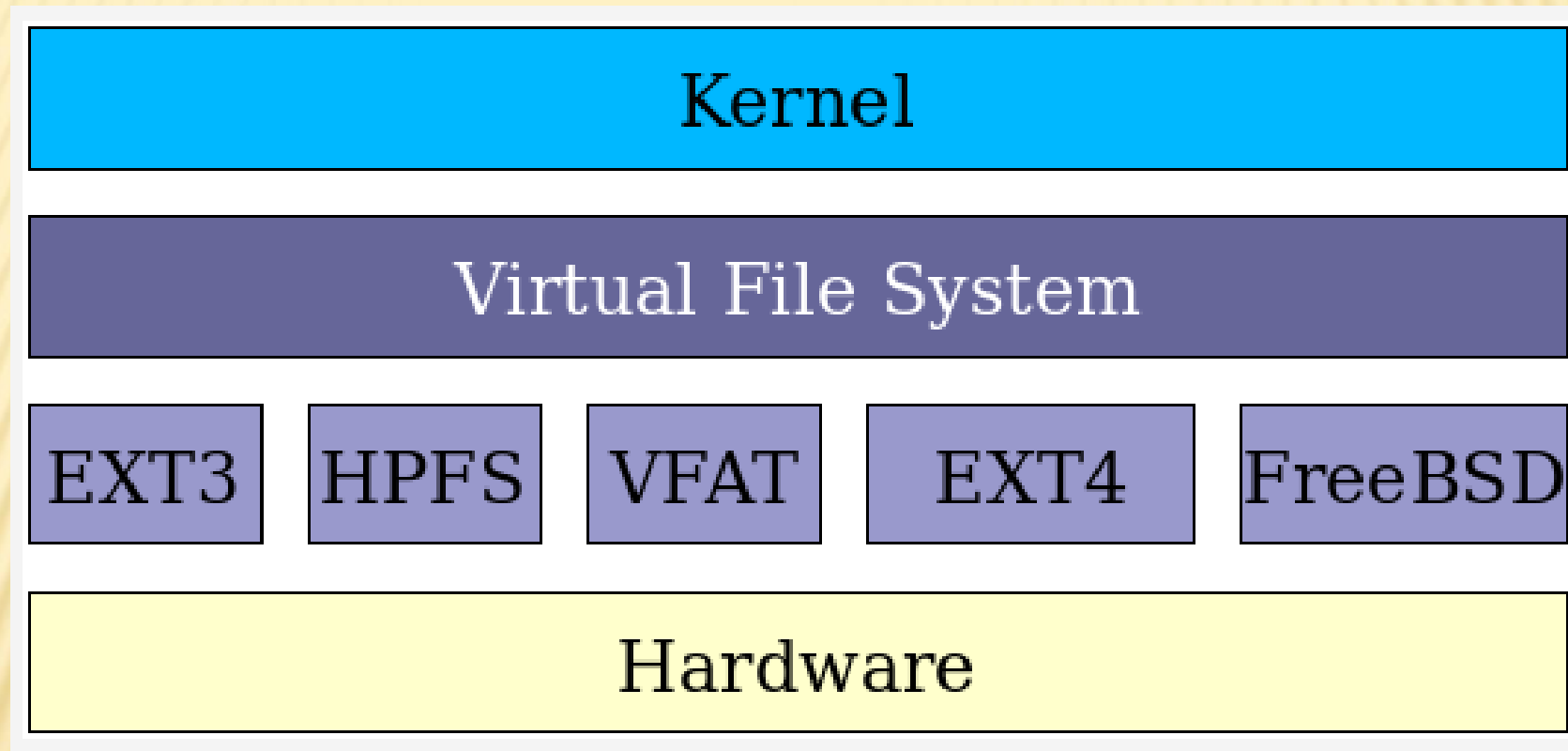
- Maximum file size : 16TB
- Maximum volume size: 1 EXA-BYTE
- Maximum NUMBER OF file: 4 BILION
- Supports:
  - 1.POSIX permissions
  - 2.File compression
  - 3.include a USERjournal entry



# ABSOLUTE PATH VS. RELATIVE PATH

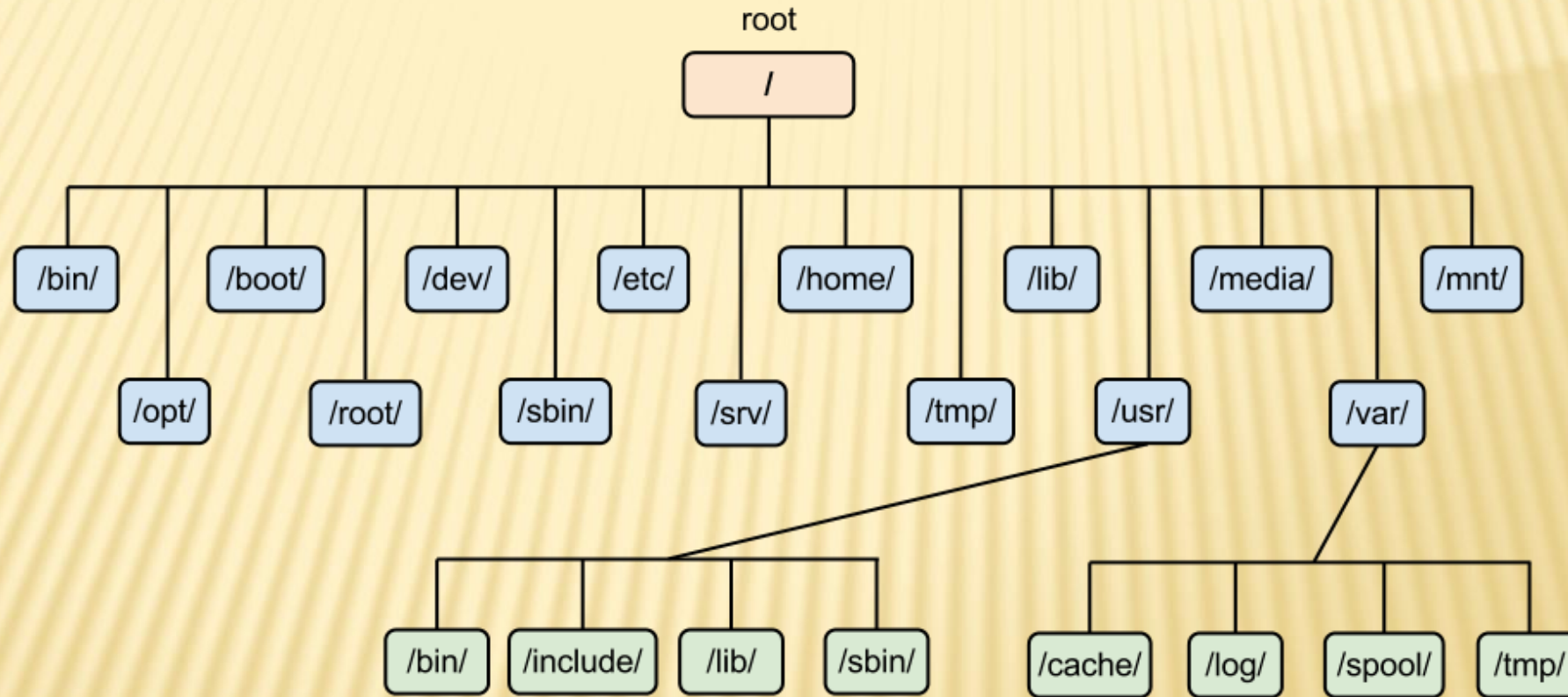


# LINUX TWO-PART FILE SYSTEM SOFTWARE



The first part of this two-part implementation is the Linux virtual file system

# ROOT STRUCTURE





# BRIEF DESCRIPTION

/ (root) filesystem	The root filesystem is the top-level directory of the filesystem. It must contain all of the files required to boot the Linux system before other filesystems are mounted. It must include all of the required executables and libraries required to boot the remaining file systems
/bin	The /bin directory contains user executable files.
/boot	Contains the static bootloader and kernel executable and configuration files required to boot a Linux computer.
/dev	This directory contains the device files for every hardware device attached to the system. These are not device drivers, rather they are files that represent each device on the computer and facilitate access to those devices.

/etc	Contains the local system configuration files for the host computer
/home	Home directory storage for user files. Each user has a subdirectory in /home.
/lib	Contains shared library files that are required to boot the system.
/media	A place to mount external removable media devices such as USB thumb drives that may be connected to the host.
/mnt	A temporary mountpoint for regular filesystems (as in not removable media) that can be used while the administrator is repairing or working on a filesystem.
/opt	Optional files such as vendor supplied application programs should be located here.



/sbin	System binary files. These are executables used for system administration.
/tmp	Temporary directory. Used by the operating system and many programs to store temporary files. Users may also store files here temporarily. Note that files stored here may be deleted at any time without prior notice.
/usr	These are shareable, read-only files, including executable binaries and libraries, man files, and other types of documentation.
/var	Variable data files are stored here. This can include things like log files, MySQL, and other database files, web server data files, email inboxes, and much more.



# BASIC COMMAND

To explore the filesystem yourself, use the `cd` command:

```
cd
```

will take you to the directory of your choice (*cd* stands for *change directory*).

```
pwd
```

will always tell you where you (*pwd* stands for *print working directory*). Also,

```
cd
```

with no options or parameters, will take you back to your own home directory, where things are safe and cosy.

# BASIC COMMAND

```
cd ..
```

will take you up one level, getting you one level closer to the / root directory. If you are in */usr/share/wallpapers* and run `cd ..`, you will move up to */usr/share*.

To see what a directory contains, use

```
ls
```

To see what a directory contain with all of hidden file

```
Ls -al or ls -a
```

# MOUNTING

The **mount** command mounts a storage device or filesystem, making it accessible and attaching it to an existing directory structure.

Syntax:

```
$ mount -t type file-system mount-point
```

added a disk /dev/sdb on /data directory

```
$ mount -t ext4 /dev/sdb /data
```



# UNMOUNTING

---

The **umount** command "unmounts" a mounted filesystem. Run `umount` command with disk name or mount point name to unmount currently mounted disk.

Syntax:

```
$ umount /dev/sdb
```

```
$ umount /data
```

# THANK YOU