Introduction to Linux and Basics of System Calls

Abhijit A. M.
abhijit.comp@coep.ac.in

Why GNU/Linux ?

Few Key Concepts

- ***Files* don't open themselves**
  - **Always some application/program open()s a file.**
- **Files don't display themselves**
  - **A file is displayed by the program which opens it. Each program has it's own way of handling ifles**

Few Key Concepts

- Programs don't run themselves
  - You click on a program, or run a command --> equivalent to request to Operating System to run it. The OS runs your program
- Users (humans) request OS to run programs, using Graphical or Command line interface
  - and programs open files

Path names

- Tree like directory structure
- Root directory called   /
- A complete path name for a file
  - /home/student/a.c
- Relative path names

  concept: every running program has a *current* working directory

  .   current directory

  .. parent directory

  ./Desktop/xyz/../p.c

A command

- Name of an executable file
  - For example: 'ls' is actually “/bin/ls”
- Command takes arguments
  - E.g. `ls /tmp/`
- Command takes options
  - E.g. `ls -a`

A command

- Command can take both arguments and options
  - E.g. ls -a /tmp/
- Options and arguments are basically argv[] of the main() of that program

Basic Navigation Commands

- pwd
- ls
  - ls -l
  - ls -l /tmp/
  - ls -l /home/student/Desktop
  - ls -l ./Desktop
  - ls -a

    \ls -F
- cd
  - cd /tmp/
  - cd
  - cd /home/student/Desktop
- notation: ~
  - cd ~
  - cd ~/Desktop
  - ls ~/Desktop

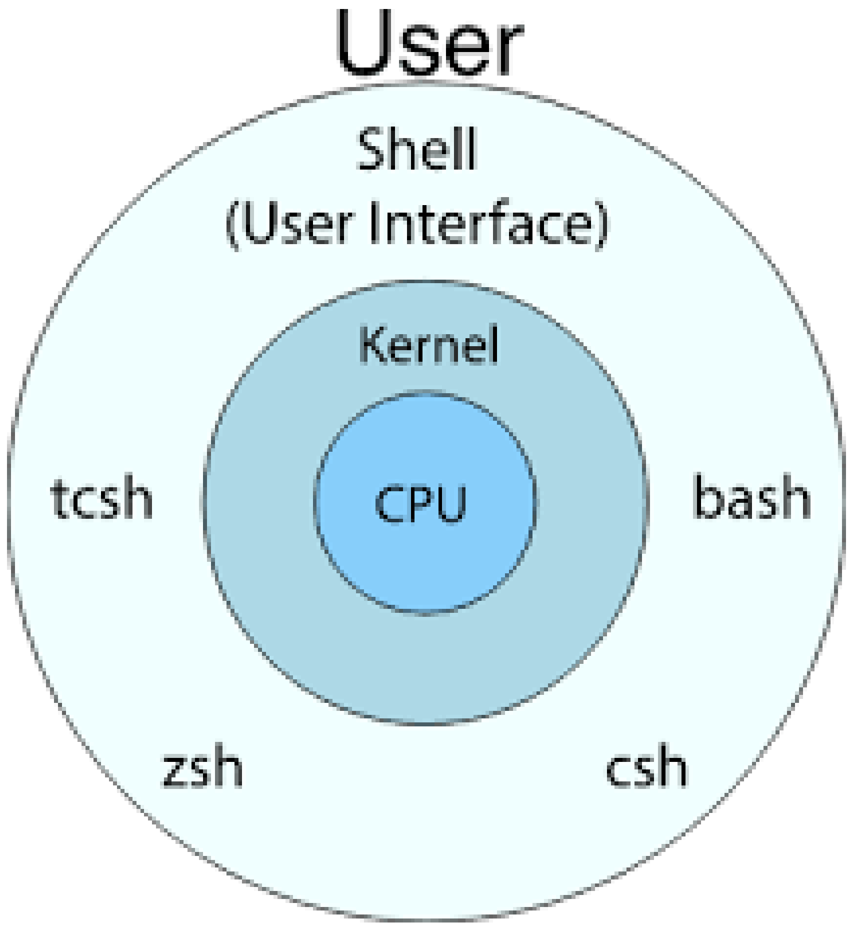
Map these commands to navigation using a graphical file browser

Before the command line, the concept of Shell and System calls

- System Call
  - A function from OS code
  - Does specific operations with hardware (e.g. reading from keyboard, writing to screen, *opening* a file from disk, etc.)
  - Applications can't access hardware directly, they have to request the OS using system calls
  - Examples
    - open(*“/x/y”, ..)*
    - *read(fd, &a, ...);*
    - *fork()*
    - *exec(“/usr/bin/ls” ...)*

The Shell

- **Shell = Cover**
- **Covers some of the Operating System's "System Calls" (mainly fork+exec) for the *Applications***
- **Talks with Users and Applications and does some talk with OS**



Not a very accurate diagram !

http://code.snipcademy.com/tutorials/linux-command-line/environment/

The Shell

Shell waits for user's input

Requests the OS to run a program which the user has asked to run

Again waits for user's input

GUI is a Shell !

Let's Understand fork() and exec()

```
#include <unistd.h>
int main() {
        fork();
        printf("hi\n");
        return 0;
}
```

```
#include <unistd.h>
int main() {
        printf("hi\n");
        execl("/bin/ls",
"ls", NULL);
        printf("bye\n");
        return 0;
}
```

A simple shell

```
#include <stdio.h>
#include <unistd.h>
int main() {
        char string[128];
        int pid;
        while(1) {
                printf("prompt>");
                scanf("%s", string);
                pid = fork();
                if(pid == 0) {
                        execl(string, string, NULL);
                } else {
                        wait(0);
                }
        }
}
```

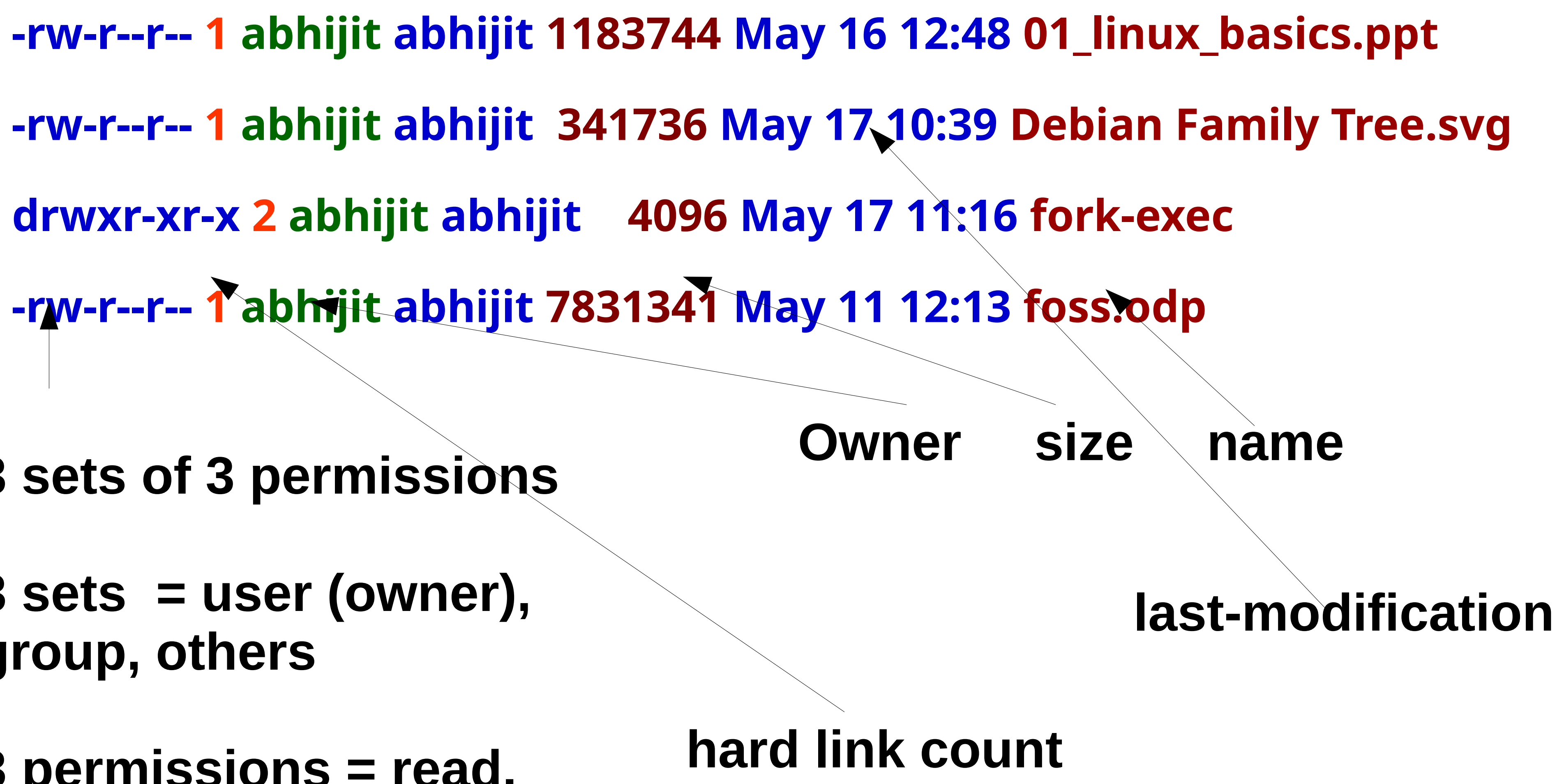
Users on Linux

- Root and others
  - root
    - superuser, can do (almost) everything
    - Uid = 0
  - Other users
    - Uid != 0
  - UID, GID understood by kernel
- Groups
  - Set of users is a group
  - Any number of groups is possible
- users/groups data in Text files: /*etc*/passwd /*etc*/shadow /*etc*/group ...

File Permissions on Linux

- 3 sets of 3 permission
  - Octal notation: Read = 4, Write = 2, Execute = 1
  - 644 means
    - Read-Write for owner, Read for Group, Read for others
- chmod command, used to change permissions, uses these notations
  - It calls the chmod() system call
- Permissions are for processes started by the user, but in common language often we say “permissions are for the user”

File Permissions on Linux



```
-rw-r--r-- 1 abhijit abhijit 1183744 May 16 12:48 01_linux_basics.ppt
-rw-r--r-- 1 abhijit abhijit  341736 May 17 10:39 Debian Family Tree.svg
drwxr-xr-x 2 abhijit abhijit    4096 May 17 11:16 fork-exec
-rw-r--r-- 1 abhijit abhijit 7831341 May 11 12:13 foss.odp
```

3 sets of 3 permissions

3 sets = user (owner), group, others

3 permissions = read, write, execute

File Permissions on Linux

- r on a file : can read the file
  - open(.... O_RDONLY) works
- w on a file: can modify the file
  - open(.... O_WRONLY) works
- x on a file: can ask the os to run the file as an executable program
  - exec(...) works
- r on a directory: can do 'ls'
- w on a directory: can add/remove files from that directory (even without 'r'!)
- x on a directory: can 'cd' to that directory

Access rights examples

- `-rw-r--r--`
  Readable and writable for file owner (actually a process started by the owner!), only readable for others
- `-rw-r-----`
  Readable and writable for file owner, only readable for users belonging to the file group.
- `drwx------`
  Directory only accessible by its owner
- `-------r-x`
  File executable by others but neither by your friends nor by yourself. Nice protections for a trap...

Permissions: more !

- Setuid/setgid bit

```
$ ls -l /usr/bin/passwd
-rwsr-xr-x 1 root root 68208 Nov 29 17:23
/usr/bin/passwd
```

- How to set the s bit?
  - chmod u+s <filename>
- What does this mean?
  - Any user can run this process, but the process itself runs as if run by the the owner of the file
    - passwd runs as if run by “root” even if you run it

Man Pages

- Manpage
  - $ man ls
  - $ man 2 mkdir
  - $ man man
  - $ man -k mkdir
- Manpage sections
  - 1 User-level cmds and apps
    - /bin/mkdir
  - 2 System calls
    - int mkdir(const char *, …);
  - 3 Library calls
    - int printf(const char *, …);
  - 4 Device drivers and network protocols
    - /dev/tty
  - 5 Standard file formats
    - /etc/hosts
  - 6 Games and demos
    - /usr/games/fortune
  - 7 Misc. files and docs
    - man 7 locale
  - 8 System admin. Cmds
    - /sbin/reboot

GNU / Linux filesystem structure

Not imposed by the system. Can vary from one system to the other, even between two GNU/Linux installations!

`/`	Root directory
`/bin/`	Basic, essential system commands
`/boot/`	Kernel images, initrd and configuration files
`/dev/`	Files representing devices
	`/dev/hda`: first IDE hard disk
`/etc/`	System and application configuration files
`/home/`	User directories
`/lib/`	Basic system shared libraries

GNU / Linux filesystem structure

`/lost+found`	Corrupt files the system tried to recover
`/media`	Mount points for removable media: `/media/usbdisk`, `/media/cdrom`
`/mnt/`	Mount points for temporarily mounted filesystems
`/opt/`	Specific tools installed by the sysadmin `/usr/local/` often used instead
`/proc/`	Access to system information `/proc/cpuinfo`, `/proc/version` ...
`/root/`	root user home directory
`/sbin/`	Administrator-only commands
`/sys/`	System and device controls (cpu frequency, device power, etc.)

GNU / Linux filesystem structure

`/tmp/` Temporary files

`/usr/` Regular user tools (not essential to the system)

`/usr/bin/`, `/usr/lib/`, `/usr/sbin`...

`/usr/local/` Specific software installed by the sysadmin (often preferred to `/opt/`)

`/var/` Data used by the system or system servers

`/var/log/`, `/var/spool/mail` (incoming mail), `/var/spool/lpd` (print jobs)...

Files: cut, copy, paste, remove,

- cat <filenames>
  - cat /etc/passwd
  - cat fork.c
  - cat <filename1> <filename2>
- cp <source> <target>
  - cp a.c b.c
  - cp a.c /tmp/
  - cp a.c /tmp/b.c
  - cp -r   ./folder1 /tmp/
  - cp -r ./folder1 /tmp/folder2
- mv <source> <target>
  - mv a.c b.c
  - mv a.c /tmp/
  - mv a.c /tmp/b.c
- rm <filename>
  - rm a.c
  - rm a.c b.c c.c
  - rm -r /tmp/a
- mkdir
  - mkdir /tmp/a /tmp/b
- rmdir
  - rmdir /tmp/a /tmp/b

Useful Commands

- echo
  - echo hi
  - echo hi there
  - echo “hi there”
  - j=5; echo $j
- sort
  - sort
  - sort < /etc/passwd
- firefox
- libreoffice
- grep
  - grep bash /etc/passwd
  - grep -i display /etc/passwd
  - egrep -i 'a|b' /etc/passwd
- less <filename>
- head <filename>
  - head -5 <filename>
  - tail -10 <filename>

Useful Commands

- **alias**

  ```
  alias ll='ls -l'
  ```

- **tar**

  ```
  tar cvf folder.tar folder
  ```

- **gzip**

  ```
  gzip  a.c
  ```

- **touch**

  ```
  touch xy.txt
  touch a.c
  ```

- **strings**

  ```
  strings a.out
  ```

- **adduser**

  ```
  sudo adduser test
  ```

- **su**

  ```
  su administrator
  ```

Useful Commands

- **df**

  ```
  df -h
  ```

- **du**

  ```
  du -hs .
  ```

- **bc**
- **time**
- **date**
- **diff**
- **wc**
- **dd**

Network Related Commands

- ifconfig
- ssh
- scp
- telnet
- ping
- w
- last
- whoami

Unix job control

- **Start a background process:**
  - gedit a.c &
  - gedit
    hit ctrl-z
    bg
- **Where did it go?**
  - jobs
  - ps
- **Terminate the job: kill it**
  - kill %*jobid*
  - kill *pid*
- **Bring it back into the foreground**
  - fg %1

Configuration Files

- **Most applications have configuration files in TEXT format**
- **Most of them are in *letc***
- ***/etc/*passwd and */etc/*shadow**
  - **Text files containing user accounts**
- ***/etc/*resolv.conf**
  - **DNS configuration**
- ***/etc/network/interfaces***
  - ***Network configuration***
- ***/etc/*hosts**
  - **Local database of Hostname-IP mappings**
- ***/etc/*apache2/apache2.conf**

~/.bashrc file

- `~/.bashrc`
  Shell script read each time a bash shell is started
- You can use this file to define
  - Your default environment variables (`PATH`, `EDITOR`...).
  - Your aliases.
  - Your prompt (see the bash manual for details).
  - A greeting message.
- Also ~/.bash_history

Mounting

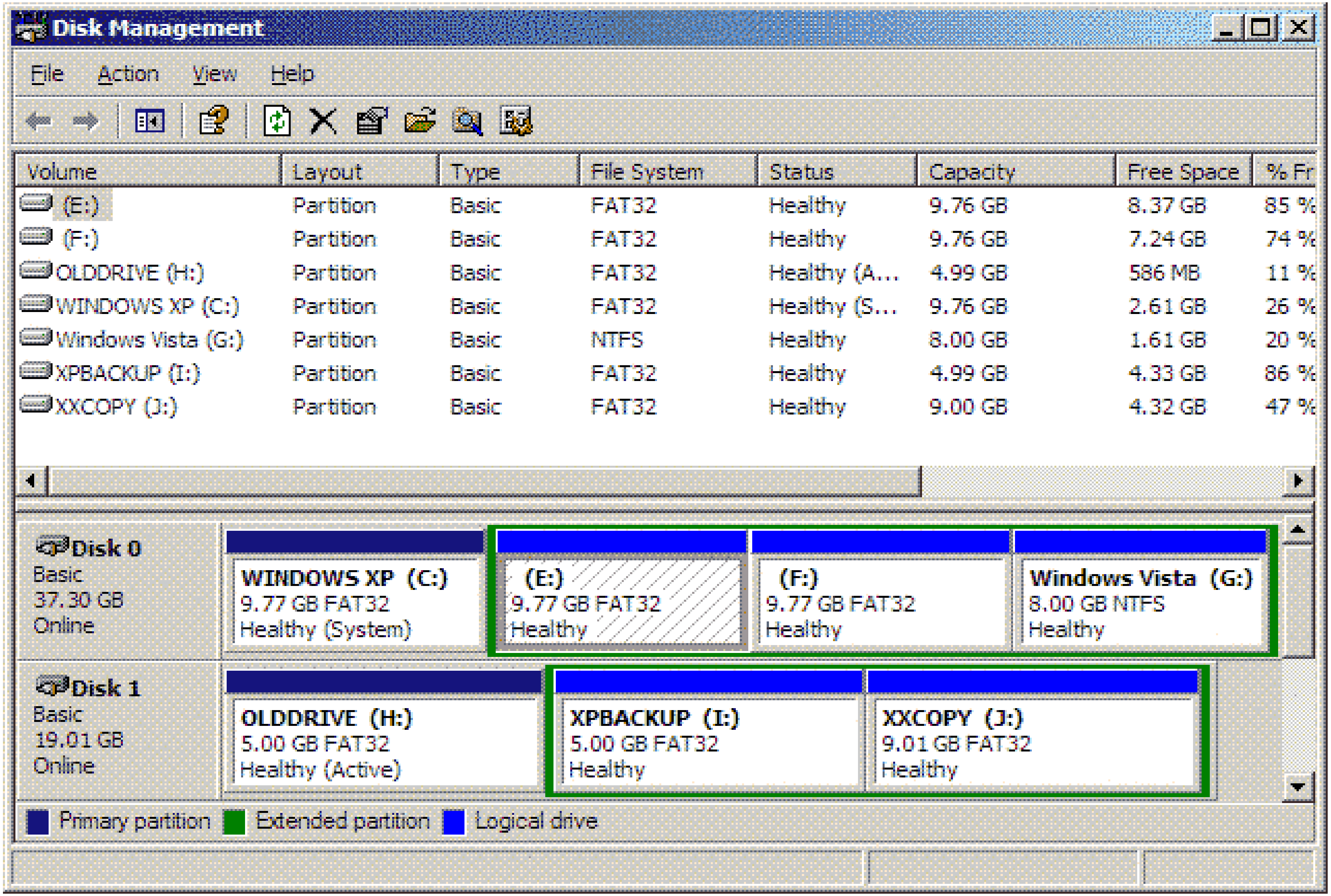
Partition

- What is C:\ , D:\, E:\ etc on your computer ?
  - "Drive" is the popular term
  - Typically one of them represents a CD/DVD RW
- What do the others represent ?
  - They are "partitions" of your "hard disk"

Partition

- **Your hard disk is one contiguous chunk of storage**
  - **Lot of times we need to "logically separate" our storage**
  - **Partition is a "logical division" of the storage**
  - **Every "drive" is a partition**
- **A logical chunk of storage is partition**
  - **Hard disk partitions (C:, D:), CD-ROM, Pen drive, …**

Partitions



Volume	Layout	Type	File System	Status	Capacity	Free Space	% Fr
(E:)	Partition	Basic	FAT32	Healthy	9.76 GB	8.37 GB	85 %
(F:)	Partition	Basic	FAT32	Healthy	9.76 GB	7.24 GB	74 %
OLDDRIVE (H:)	Partition	Basic	FAT32	Healthy (A...	4.99 GB	586 MB	11 %
WINDOWS XP (C:)	Partition	Basic	FAT32	Healthy (S...	9.76 GB	2.61 GB	26 %
Windows Vista (G:)	Partition	Basic	NTFS	Healthy	8.00 GB	1.61 GB	20 %
XPBACKUP (I:)	Partition	Basic	FAT32	Healthy	4.99 GB	4.33 GB	86 %
XXCOPY (J:)	Partition	Basic	FAT32	Healthy	9.00 GB	4.32 GB	47 %

Disk 0
Basic
37.30 GB
Online

WINDOWS XP (C:)
9.77 GB FAT32
Healthy (System)

(E:)
9.77 GB FAT32
Healthy

(F:)
9.77 GB FAT32
Healthy

Windows Vista (G:)
8.00 GB NTFS
Healthy

Disk 1
Basic
19.01 GB
Online

OLDDRIVE (H:)
5.00 GB FAT32
Healthy (Active)

XPBACKUP (I:)
5.00 GB FAT32
Healthy

XXCOPY (J:)
9.01 GB FAT32
Healthy

■ Primary partition 🟩 Extended partition 🟦 Logical drive

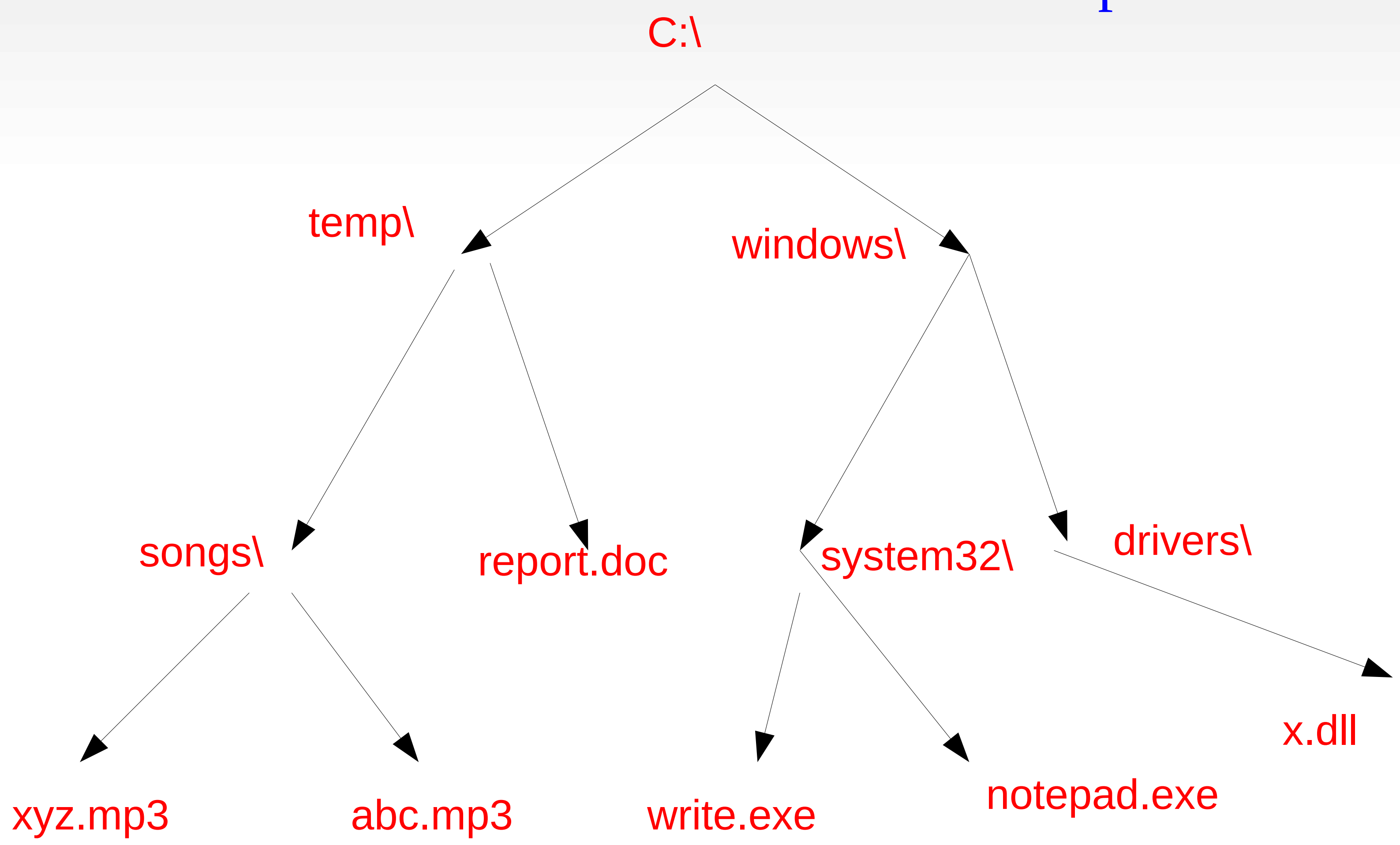
Managing partitions and hard drives

- System → Administration → Disk Utility
- Use gparted or fdisk to partition drives on Linux
- Had drive partition names on Linux
  - /dev/sda → Entire hard drive
  - /dev/sda1, /dev/sda2, /dev/sda3, ... Different partitions of the hard drive
  - Each partition has a *type* – ext4, ext3, ntfs, fat32, etc.
- Formatting: creating an empty layout on disk, layout capable of storing the tree of files/folders
  - There are different layouts named ext4, ext2, ntfs, etc.

Windows Namespace

c:\temp\songs\xyz.mp3

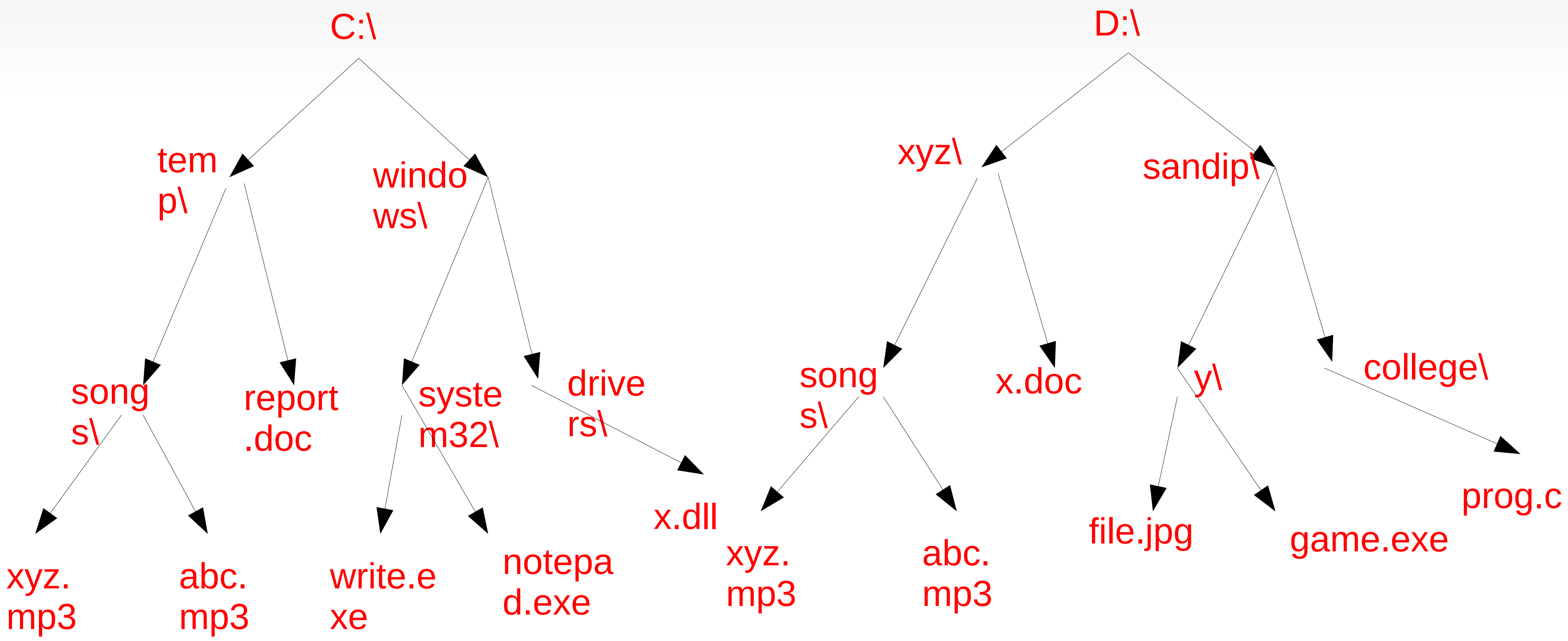
- Root is C:\ or D:\ etc
- Separator is also “\”



Windows Namespace

- C:\ D:\ Are partitions of the disk drive
- Typical convention: C: contains programs, D: contains data

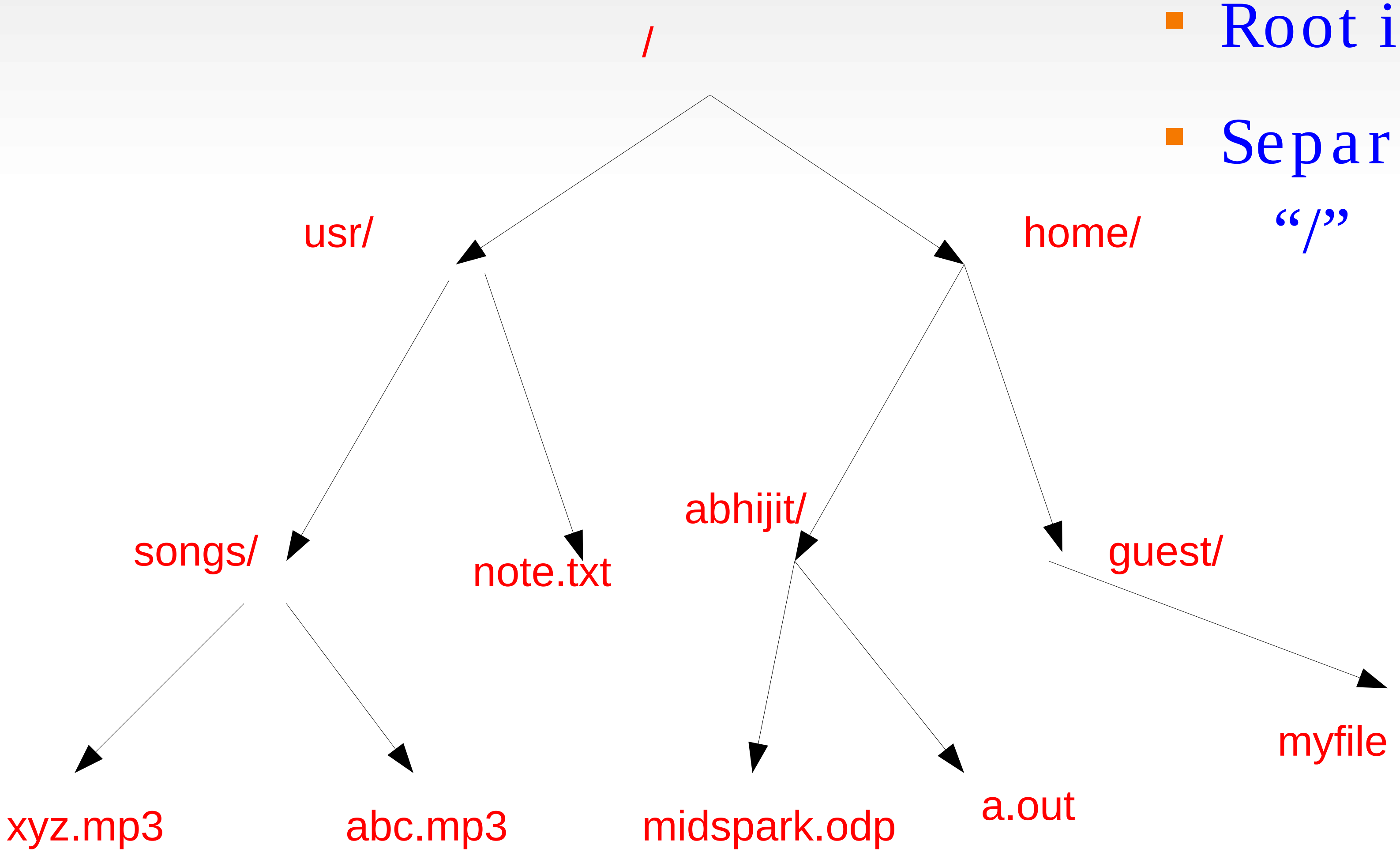
- One "tree" per partition
  - Together they make a "forest"



Linux Namespace: On a partition

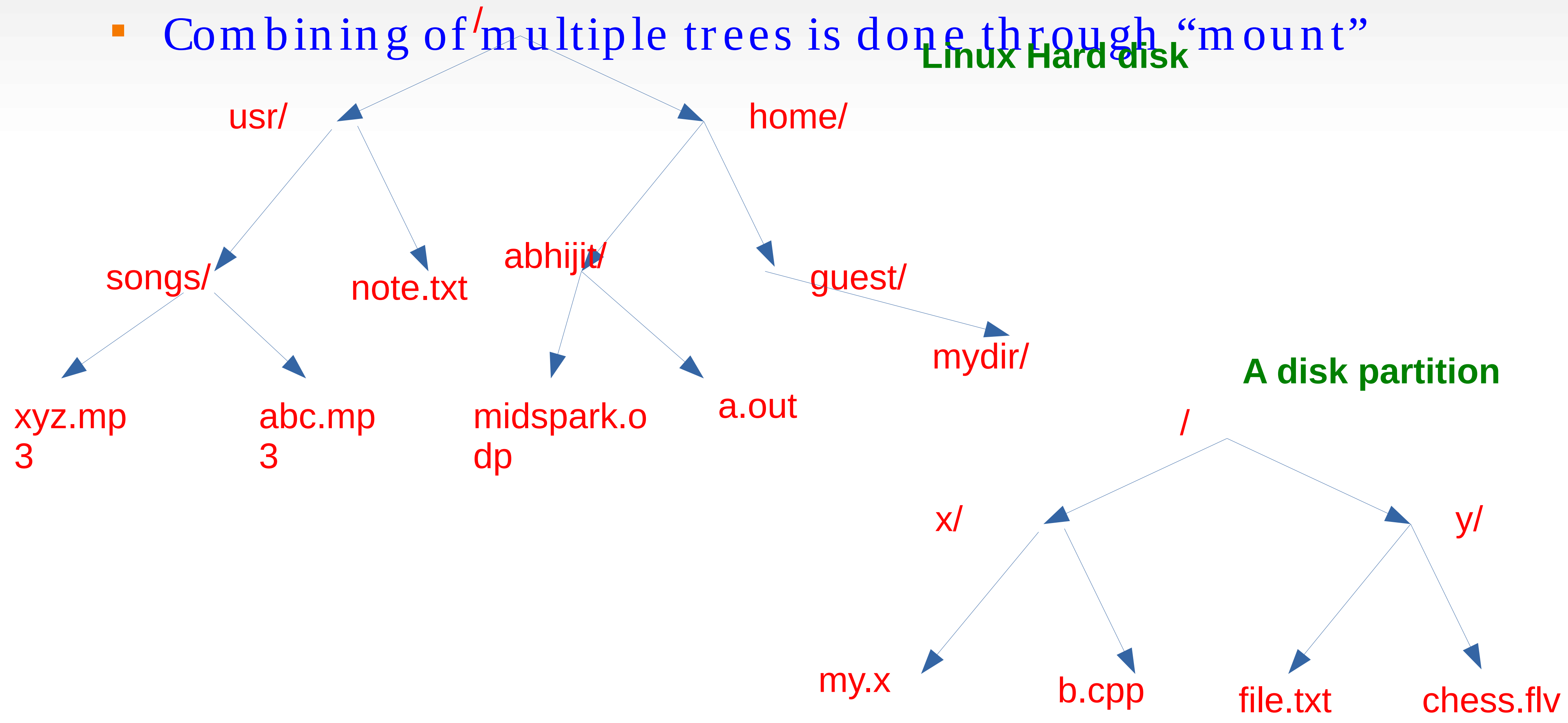
/usr/songs/xyz.mp3

- On every partition:
  - Root is "/"
  - Separator is also "/"

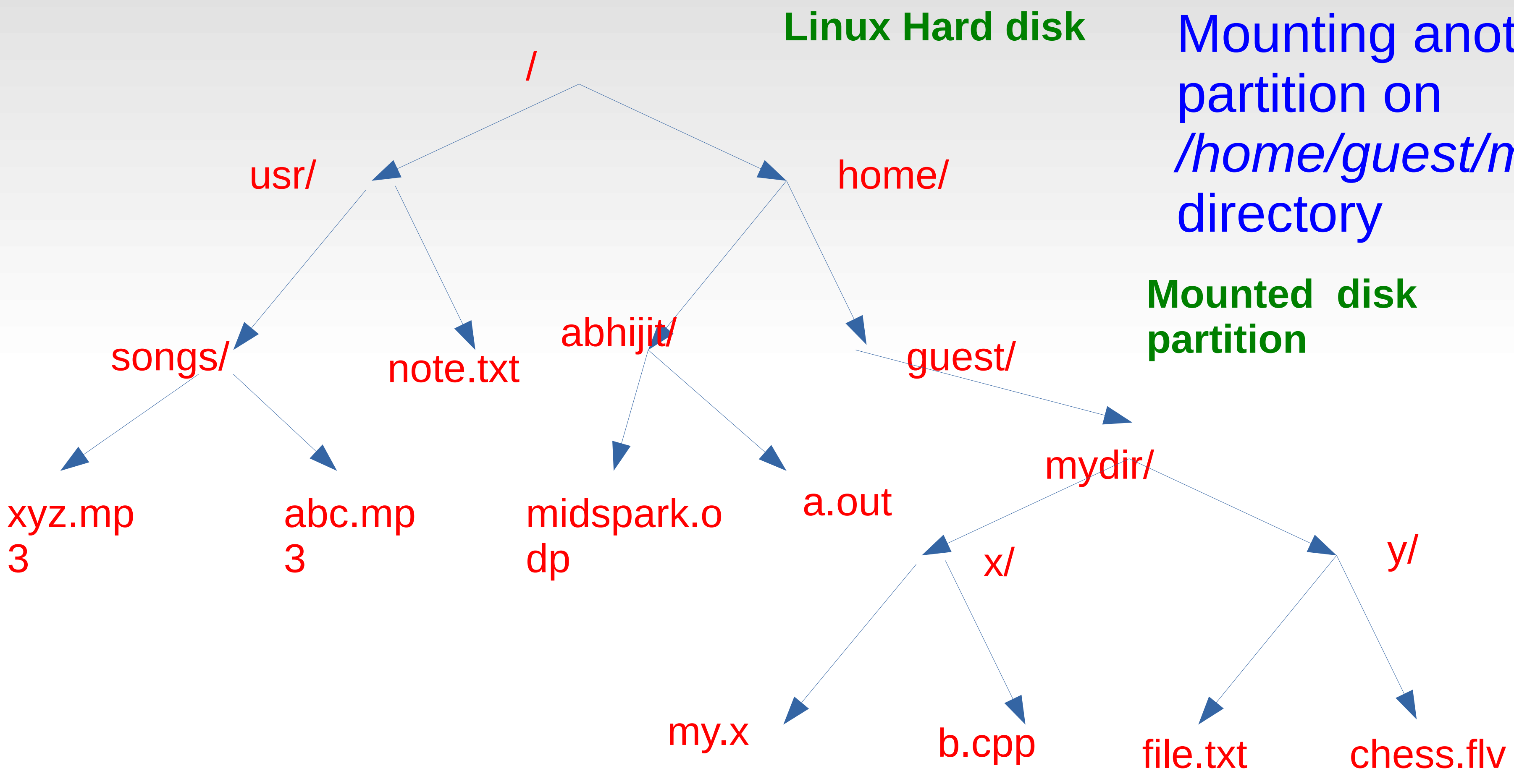


Linux namespace: Mount

- Linux namespace is a single "tree" and not a "forest" like Windows
- Combining of multiple trees is done through "mount"



Linux namespace
Mounting a partition



Mounting another partition on */home/guest/mydir/* directory

/home/guest/mydir/x/b.cpp → way to access the file on the other disk partition

Mounting across network!

Using Network File System (NFS)

```
sudo apt install nfs-common
```

```
$ sudo mount 172.16.1.75:/mnt/data  /myfolder
```

Files that are not regular/directory

Special devices (1)

Device files with a special behavior or contents

- `/dev/null`
  The data sink! Discards all data written to this file.
  Useful to get rid of unwanted output, typically log information:
  ```
  mplayer black_adder_4th.avi &> /dev/null
  ```
- `/dev/zero`
  Reads from this file always return `\0` characters
  Useful to create a file filled with zeros:
  ```
  dd if=/dev/zero of=disk.img bs=1k count=2048
  ```

See `man null` or `man zero` for details

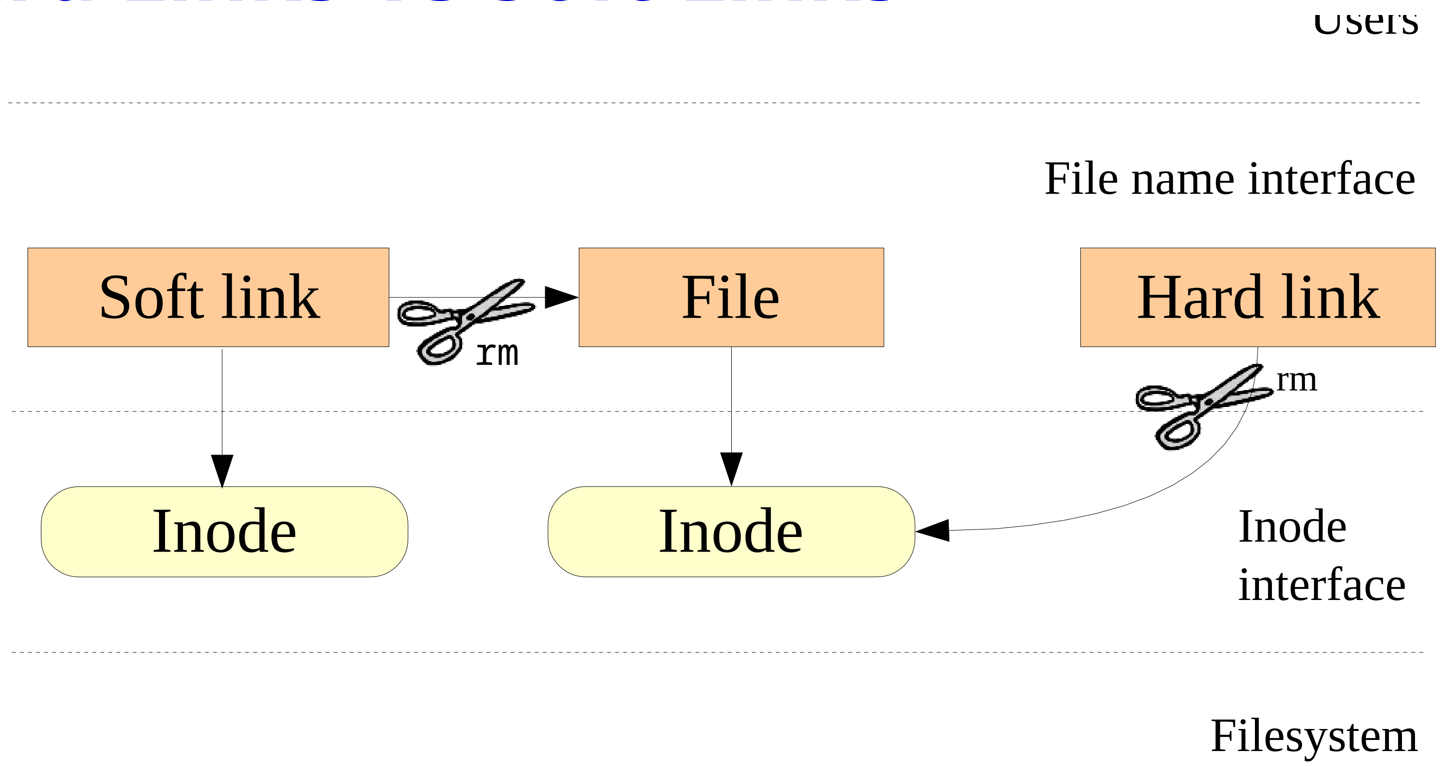
Special devices (2)

- `/dev/random`
  Returns random bytes when read. Mainly used by cryptographic programs. Uses interrupts from some device drivers as sources of true randomness ("entropy").
  Reads can be blocked until enough entropy is gathered.

- `/dev/urandom`
  For programs for which pseudo random numbers are fine.
  Always generates random bytes, even if not enough entropy is available (in which case it is possible, though still difficult, to predict future byte sequences from past ones).

Files names and inodes

Hard Links Vs Soft Links



Creating “links”

- Hard link

```
$ touch m
$ ls -l m
-rw-rw-r-- 1 abhijit abhijit 0 Jan  5 16:18 m
$ ln m mm
$ ls -l m mm
-rw-rw-r-- 2 abhijit abhijit 0 Jan  5 16:18 m
-rw-rw-r-- 2 abhijit abhijit 0 Jan  5 16:18 mm
$ ln mm mmm
$ ls -l m mm mmm
-rw-rw-r-- 3 abhijit abhijit 0 Jan  5 16:18 m
-rw-rw-r-- 3 abhijit abhijit 0 Jan  5 16:18 mm
-rw-rw-r-- 3 abhijit abhijit 0 Jan  5 16:18 mmm
$ echo Hi > m
$ ls -l m mm mmm
-rw-rw-r-- 3 abhijit abhijit 3 Jan  5 16:18 m
-rw-rw-r-- 3 abhijit abhijit 3 Jan  5 16:18 mm
-rw-rw-r-- 3 abhijit abhijit 3 Jan  5 16:18 mmm
$ cat m
Hi
$ cat mm
Hi
```

- Soft Link