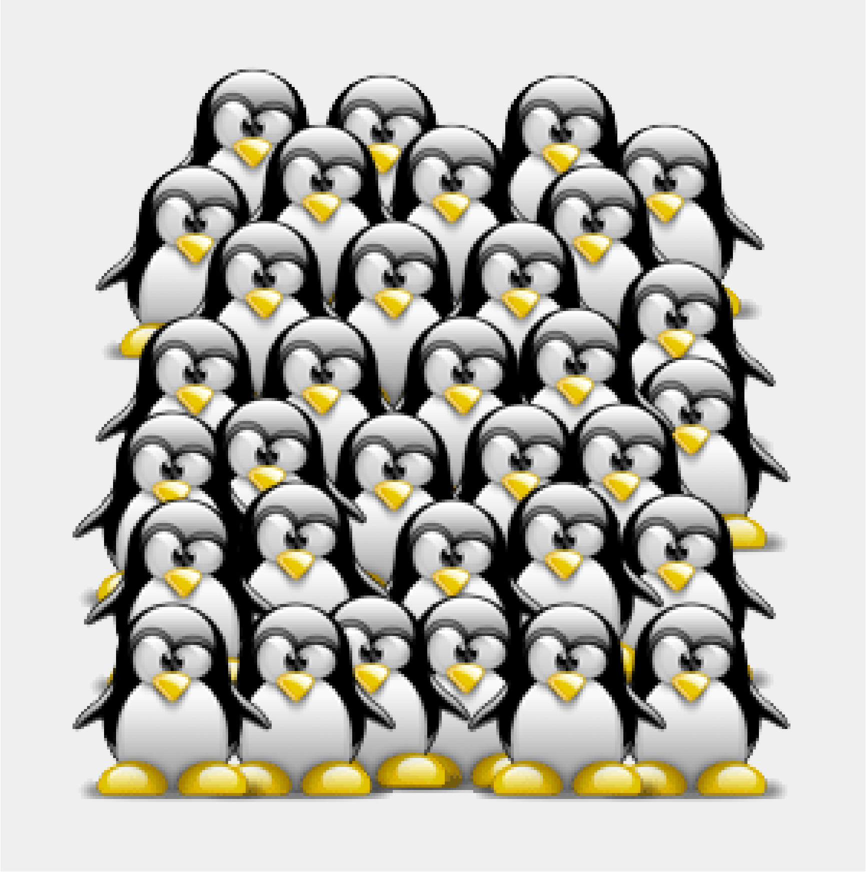
GNU/Linux Course
Lesson 3

Puria Nafisi Azizi @pna ✓
http://netstudent.polito.it

- ***User management***
- Passwords
- Remote access
- Permissions



User management

Unix has permission from the earliest versions, for different users to work simultaneously on the system, as well as GNU/Linux is a multiuser system, so the permissions and restrictions on individual users and groups are much more stringent than on other proprietary systems for as households and very similar to what is coded for enterprise-class systems.

In brief, a user's data can not be edited and viewed by other users except in the case that this is explicitly permitted (obviously the root user has no limitations).

User management

Even the operations you can perform on a GNU/Linux are subject to this ACL system, which effectively makes it even more difficult the creation and spread of viruses on GNU/Linux.

To access a GNU/Linux, except in rare cases, you must log into the system by entering your username and password.

ACL stands for... -> search on wikipedia

User management

adduser / ***useradd***: When invoked without the *-D* option, the *useradd* command creates a new user account using the values specified on the command line and defaults by the system. The new user account is added to the system files that need it, you will create the home directory, and there you will copy the original file, depending on the options on the command line.

The main options that apply to the ***useradd*** command are

-d *home_dir*

The new user is created using *home_dir* as the value for the user's login directory.

User management

-e *expiration_date*

The time off dell''account user (MM/DD/YY).

-f *days_before_expire*

The number of days after password expires until the account is permanently disabled. 0 disables the account immediately, -1 to disable this feature.

-g *initial_group*

User group name or GID, which must exist.

User management

-G group,[...]

Other groups the user belongs. Must be entered separated by commas, no spaces.

-s shell

The name of the user's login shell.

User management

-u uid

The numerical value of the user. This value must be unique, unless you used the **-o** option. The value must be non-negative. The default behavior is to use the smallest ID value greater than 999 and greater than that of any other person. Values between 0 and 999 are typically reserved for system accounts.

File Reference:

/etc/passwd - user account information

/etc/default/useradd - default information

/etc/skel - directory containing default files

User management

passwd: *Modify the login password of a user*

```
passwd [user [password]]
```

The new password must be at least six characters long and can consist of both uppercase and lowercase letters or non-alphabetic characters. Obviously you can not use your name as a password.

When using the command by root, the rules for passwords are not enforced and does not require the old password in advance.

- User management
- ***Passwords***
- Remote access
- Permissions

Passwords

/etc/passwd contains the entire database of users of the system, enclosed in a plain text file. It is sometimes also referred to as the password file, although in practice, with the introduction of ***/etc/shadow*** file that no longer contains the password.

1. Username
2. Password in encrypted form, or in the presence of x /etc/shadow
3. Numeric user ID
4. Group ID number
5. Full name or description of account
6. Home directory
7. Login shell

Password

The file **/etc/passwd** is readable by all users of the system, so that the search for someone else, but this also means that a system with no **/etc/shadow** in this file also contains the password, even if they are in encrypted form.

Passwords

/etc/group contains all the groups database enabled on the system in the form of text files and has a very simple structure of the record:

group: password: GID: user list

- ***group***

This is the name that identifies the group, along with its GID.

- ***password***

If used, the group password in this file is stored in encrypted form, and there are any restrictions for this file in ***/etc/passwd.*** The use of group passwords is not very widespread.

Password

- ***GID***

It is the identification number of the group is the value entered in /etc/passwd to specify the default group of a user. A user can belong to several distinct groups of course, but can have only one default group.

- ***user list***

The set of all users in a group.

Passwords

Unfortunately, the encrypted passwords in **/etc /passwd**, especially in the presence of weak passwords, or without reasonable safety criteria (too short or trivial) can be decoded in a short time with modern systems.

From here the need to store passwords in another file, /etc/shadow, in fact, only readable by root and in which passwords are still stored in encrypted form: no administrator should be able to know user passwords.

For added security, finally, only the authentication programs and management may have access to **/etc/shadow**, that looks like this:

username:password:lastchange:min:max:warn:inactive:expire:

Passwords

username: user name.

password: encrypted password or * (user disabled) and! (no password).

lastchange: number of days since last password change (in a Unix, so from January 1, 1970).

min: minimum days before they can change the password.

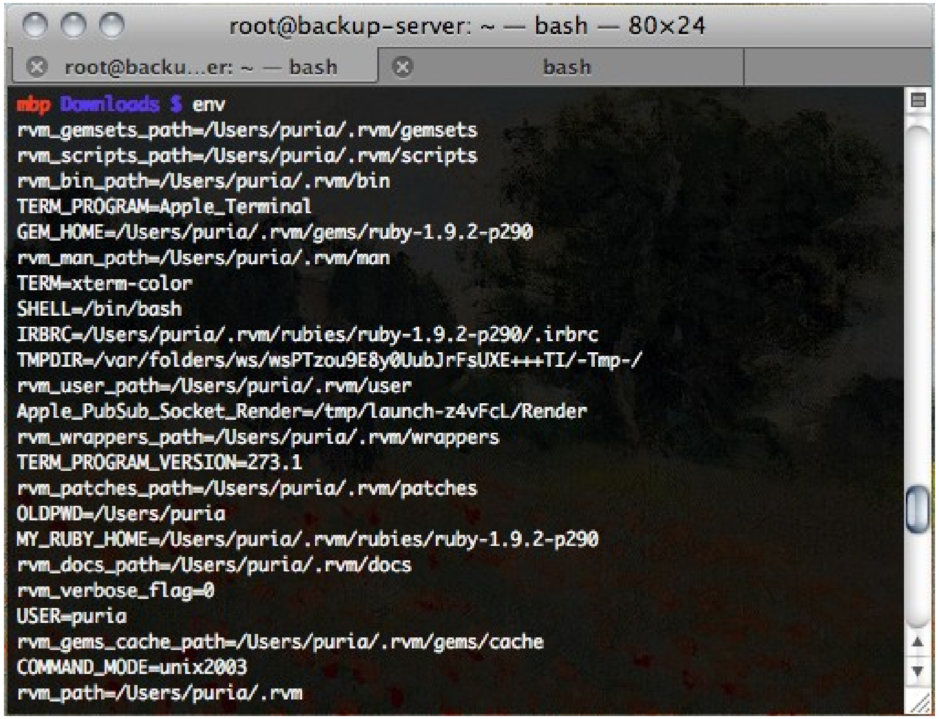
max: maximum password age (always in days);

warn: number of days notice for the password.

inactive: the number of days of inactivity the user is granted

expire: Date of disabling the account.

- The Environment
- Jobs (not Steve)
- ***Filesystem structure***


root@backup-server: ~ — bash — 80×24
root@backu...er: ~ — bash
bash

```
mbp Downloads $ env
rvm_gemsets_path=/Users/puria/.rvm/gemsets
rvm_scripts_path=/Users/puria/.rvm/scripts
rvm_bin_path=/Users/puria/.rvm/bin
TERM_PROGRAM=Apple_Terminal
GEM_HOME=/Users/puria/.rvm/gems/ruby-1.9.2-p290
rvm_man_path=/Users/puria/.rvm/man
TERM=xterm-color
SHELL=/bin/bash
IRBRC=/Users/puria/.rvm/rubies/ruby-1.9.2-p290/.irbrc
TMPDIR=/var/folders/ws/wsPTzou9E8y0UubJrFsUXE+++TI/-Tmp-/
rvm_user_path=/Users/puria/.rvm/user
Apple_PubSub_Socket_Render=/tmp/launch-z4vFcL/Render
rvm_wrappers_path=/Users/puria/.rvm/wrappers
TERM_PROGRAM_VERSION=273.1
rvm_patches_path=/Users/puria/.rvm/patches
OLDPWD=/Users/puria
MY_RUBY_HOME=/Users/puria/.rvm/rubies/ruby-1.9.2-p290
rvm_docs_path=/Users/puria/.rvm/docs
rvm_verbose_flag=0
USER=puria
rvm_gems_cache_path=/Users/puria/.rvm/gems/cache
COMMAND_MODE=unix2003
rvm_path=/Users/puria/.rvm
```

Filesystem structure

The computer disk is divided into one or more data containers, called partitions. Each *partition* is organized according to a *filesystem*, which determines the way in which data is written to disk, and how they are reported to the user.

Within a filesystem data are usually organized into *files and directories*.

Common filesystems are:
EXT2/EXT3/EXT4/ReiserFs/XFS/JFS (GNU / Linux), FAT / NTFS (Windows), HFS/HFS+ (Mac OS), UFS (BSD).

Filesystem structure

DOGMA:

"*In Linux, everything is represented as a file*"

(except very few and rare exceptions)

A file is an abstraction for a '*something*':

a place to keep data (documents, binary)

System of a physical device (mouse, various adapters)

abstractions for communication (a pipe, a socket)

a link to another file content for other files (ie a directory)

Filesystem structure

The filenames have a maximum length (usually 256 characters) and you can use all the characters (even if it is not recommended to use special ones), except the separator '/'.

The file names are case sensitive, so a lowercase letter is not *equivalent* to a capital letter.

All files have *permissions*, which define the tasks allowed by each user on each file.

Filesystem structure

DLE directory in a filesystem are organized in *a tree diagram*. The root is denoted by '/' and is called 'root' (*not to be confused with the super user*).

There will be no different trees for each partition, as the directory tree is unique for the whole system

Filesystem structure

However, the tree of a GNU/Linux allows the integration of different file systems from many records (fixed and removable).

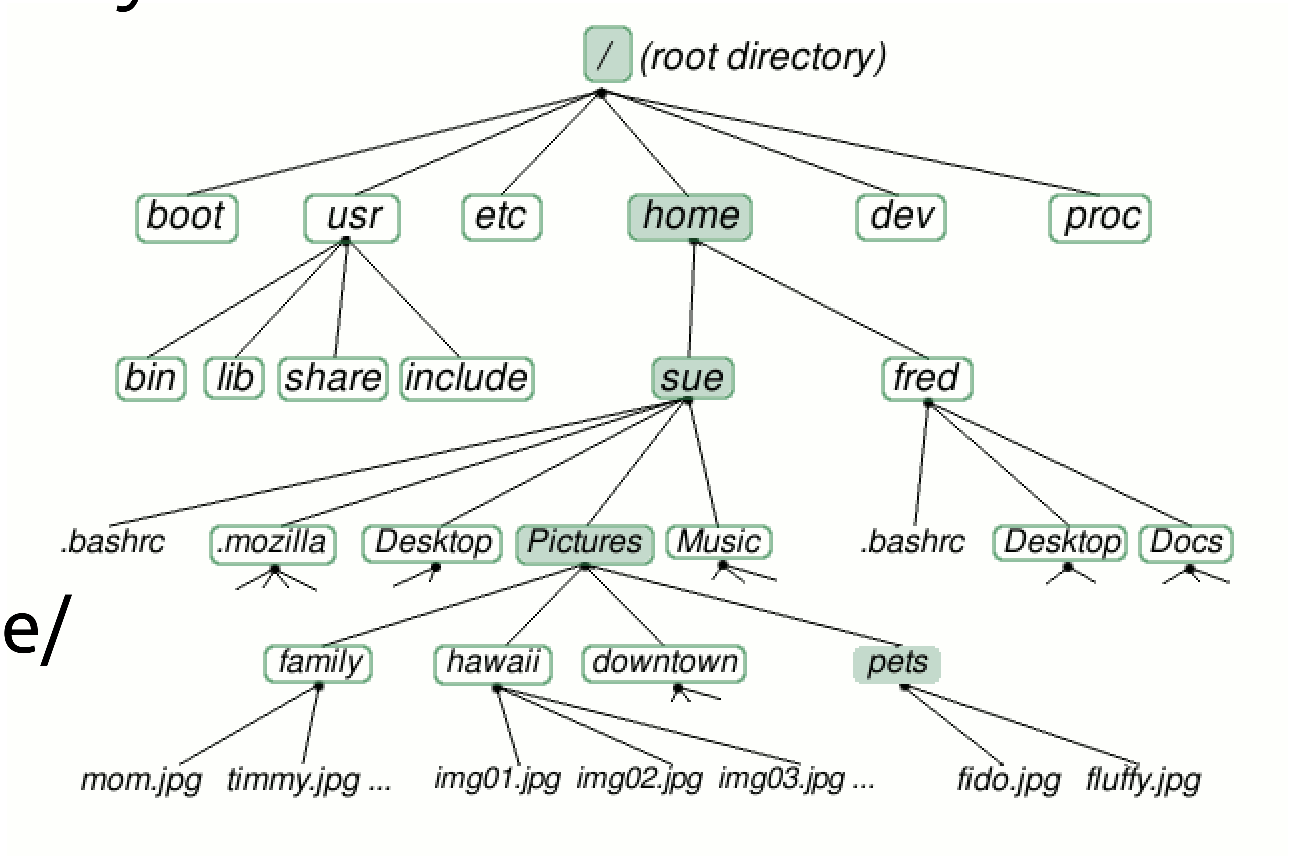
The operation of ***mount*** provide access to **a file** in a certain **position of the tree** (*called the mount point*).

It is also possible to mount file systems on other computers connected via the network

Filesystem structure

An example of hierarchy is:

- /
  - /dev
  - /home
  - /proc
  - /usr
    - /usr/share/
  - /root
  - /var
  - /tmp



Filesystem structure

'/' is the root, all other directories or files are descended from it:

'**/bin**' contains the executable files of many basic commands

'**/boot**' contains the files of the kernel and boot image, in addition to LILO and Grub. It is often advisable that the directory resides in a partition at the beginning of the disc.

Filesystem structure

- **'/dev'** only contains special files, including those relating to the devices. These are virtual files are not physically on the disk:)
- Some interesting examples are:
- The file **'/dev/**null' which can be sent to destroy any file or string
- The file **'/dev/zero'**, which contains an infinite sequence of 0
- The file **'/dev/random'**which contains an infinite sequence of random values
- The file **'/dev/hda' or '/dev/sda'** (for example) contain the entire disk

Filesystem structure

- **'/proc'** contains several files containing information about the system, kernel and processes (also not physically present on the disc)
- in **'/usr'** go all the executable files, libraries, source of most of the *system* programs. For this reason, most of the files contained therein is read-only (for the normal user)
- **'/usr/bin'** contains basic user commands

Filesystem structure

- **'/usr/sbin'** contains additional commands for the administrator
- **'/usr/lib'** contains the system libraries
- **'/usr/share'** contains documentation or common to all libraries, for example **'/usr/share/man'** contains the text of the manpage
- **'/var'** contains files usually written by the kernel services, such as log

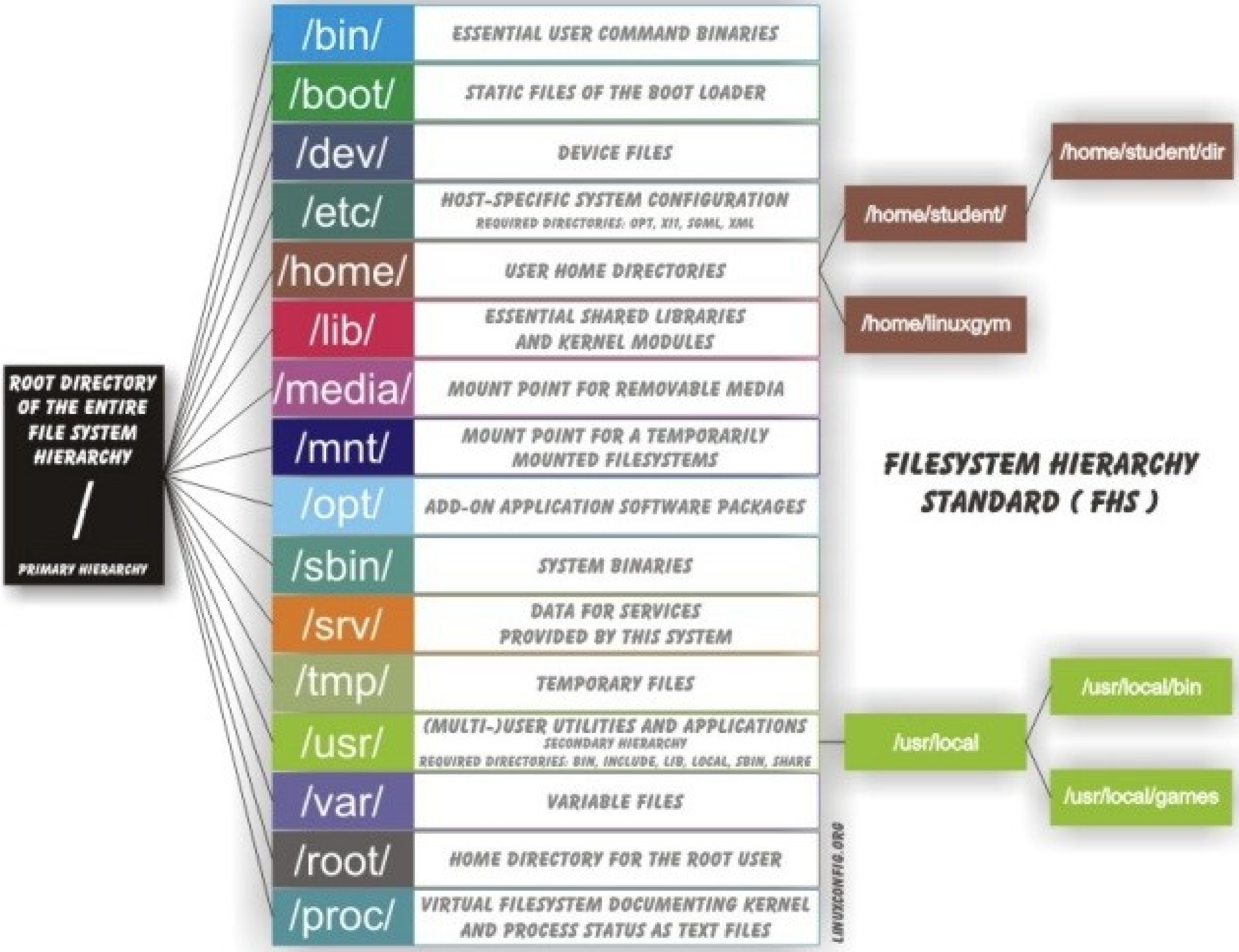
Filesystem structure

- **'/etc'** contains the configuration files of the system, primarily in reading and writing by the administrator and services, such as the password file
- **'/home'** directory contains the user's home system
- **'/mnt' and '/media'** is the directory where you place the file added
- **'/opt'** need for some additional applications

Filesystem structure

- **'/tmp'** is a temporary directory writable
- **'/root'** is the directory user account
-
- These directories are usually all present immediately after the installation of a GNU / Linux.

Filesystem structure



/bin/	ESSENTIAL USER COMMAND BINARIES
/boot/	STATIC FILES OF THE BOOT LOADER
/dev/	DEVICE FILES
/etc/	HOST-SPECIFIC SYSTEM CONFIGURATION REQUIRED DIRECTORIES: OPT, X11, SGML, XML
/home/	USER HOME DIRECTORIES
/lib/	ESSENTIAL SHARED LIBRARIES AND KERNEL MODULES
/media/	MOUNT POINT FOR REMOVABLE MEDIA
/mnt/	MOUNT POINT FOR A TEMPORARILY MOUNTED FILESYSTEMS
/opt/	ADD-ON APPLICATION SOFTWARE PACKAGES
/sbin/	SYSTEM BINARIES
/srv/	DATA FOR SERVICES PROVIDED BY THIS SYSTEM
/tmp/	TEMPORARY FILES
/usr/	(MULTI-)USER UTILITIES AND APPLICATIONS SECONDARY HIERARCHY REQUIRED DIRECTORIES: BIN, INCLUDE, LIB, LOCAL, SBIN, SHARE
/var/	VARIABLE FILES
/root/	HOME DIRECTORY FOR THE ROOT USER
/proc/	VIRTUAL FILESYSTEM DOCUMENTING KERNEL AND PROCESS STATUS AS TEXT FILES

Filesystem structure

The connections are managed with the command '**ln**'.

They can be of two types, depending on their implementation in the filesystem: *soft and hard*.

Hard links allow you to access a file on the disk by two different paths, they are rarely used, and do not allow links between file systems of two different partitions.

Filesystem structure

Soft links are created with the command

- ln -s $ SOURCE $ DESTINATION

The shortcut file is created a pointer (at the filesystem level) to the source file.

It occupies very little space and is indicated by 'the beginning of the string of permits.

Usually equal access to soft link to access the file destination.