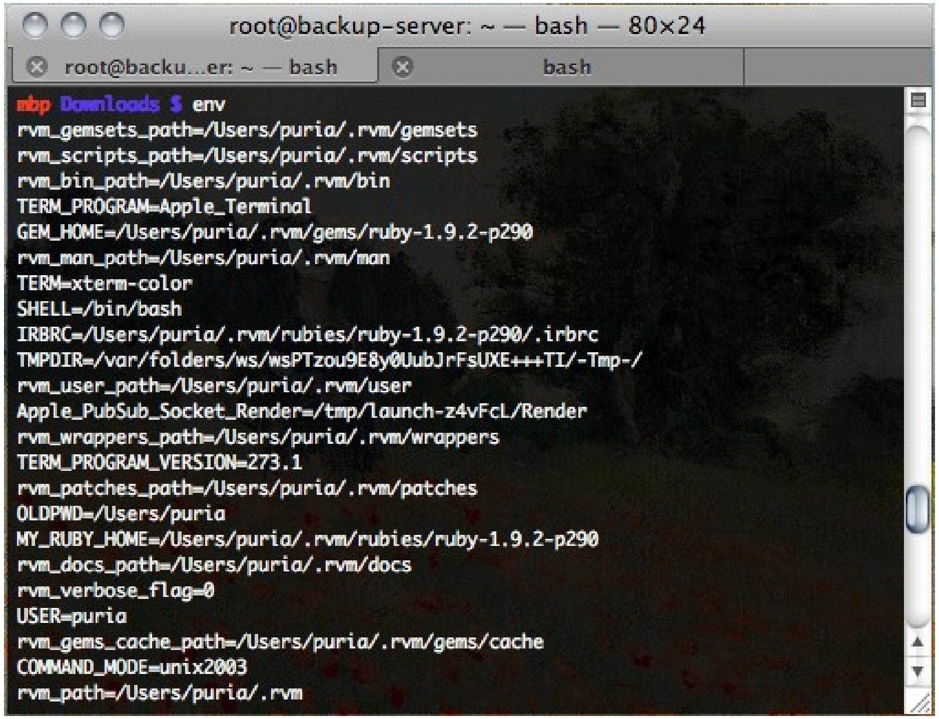
GNU/Linux Course
Lesson 2

Puria Nafisi Azizi @pna ✓

http://netstudent.polito.it

- ***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
```

The environment

In the shell there are a number of variables preset.

You access the list of these with the command ***env***

For example:

$HOME - your home directory

$SHELL - the current shell

$PATH - list of directories that reside programs

The environment

Some commands provide basic information about the status of the system:

uptime - system load and the last reboot

uname - kernel currently running

free - total memory used by the system, it is the main (RAM) that the additional written to disk (swap)

vmstat - like free, more detailed

The environment

Other environment variables can be set using the command ***export***, for example:

```
export HOME = "18:30"← set the variable
echo $HOME  ← prints the variable
```

It is often useful to add a location from which to run the applications:

```
export PATH = "$ PATH:/usr/local/bin"
```

The environment

You can define your own commands (alias) to the shell using the command ***alias***, for example:

removeeverything alias = 'rm -r'

alias ls = 'uptime'

With no arguments lists the currently defined aliases.

Please note that the command ***alias*** has priority over the basic commands, for example with the second alias, by running ***ls*** will launch instead of ***uptime***.

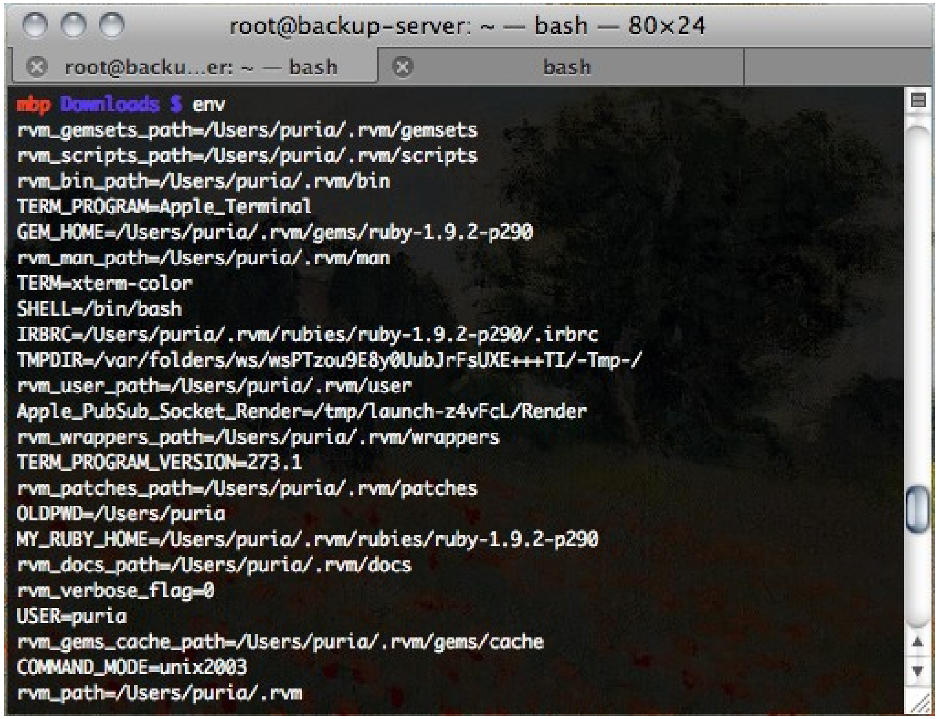
The environment

Sometimes you want to get more information in real-time system, such as allocation of memory usage and processor load, as well as knowing all the active processes (details in next slides).

To get an immediate idea of what is going on GNU/Linux, it is useful to the command ***top*** that assumes the duties of monitors in real time.

An evolution, not always installed, the command ***htop***.

- 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
```

Jobs

In computing, we define *a process currently executing a program*. The processes are referred to by their PID, a unique identifier of the process.

The ***ps*** command allows you to list all programs running on your system:

ps -ef - lists all active processes

ps faux - lists processes (organized as a tree)

The command ***pstree*** lists (tree) current processes starting from init, *the first system process* (PID = 1).

Jobs

You can influence the flow of a program by sending *the signals*. The default signal is **SIGTERM**, which indicates to the program *termination request*. **SIGKILL**, by contrast, shows the *forced termination* of a process.

The command ***kill*** is used to send a signal to a running process, which must be indicated by its PID:

the

kill -SIGTERM 1234

Similarly, the command ***killall*** allows you to terminate the execution of a program given its name (not that there is only one corresponding process).

Jobs

The natural end for all processes is the shutdown. The standard command to perform a system shutdown is ***shutdown*** , but there are several variants:

shutdown -h now <- switches off the system *now*

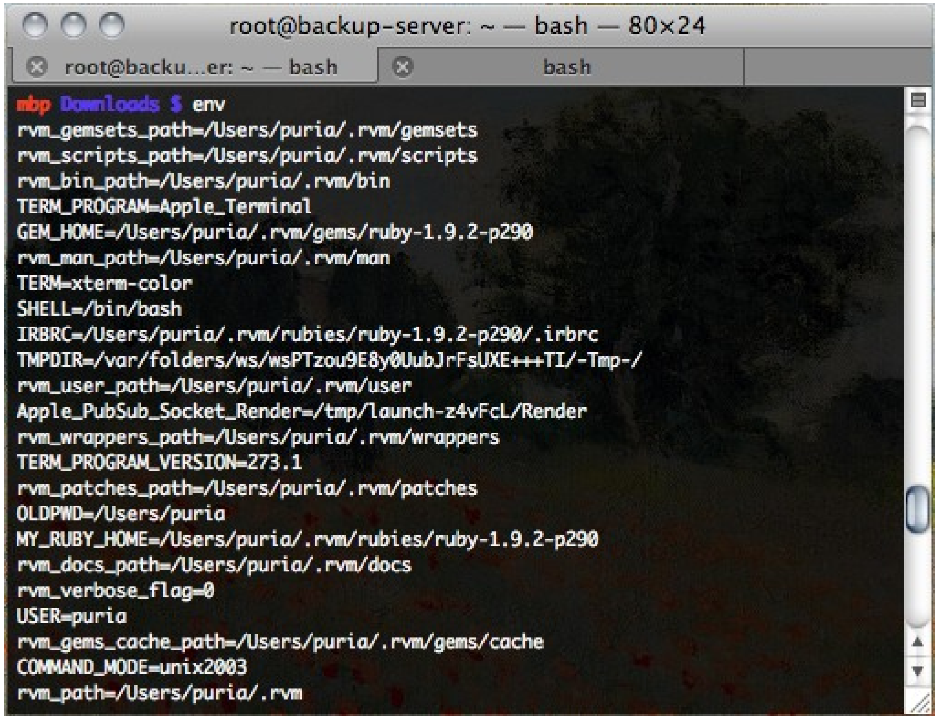
shutdown -r now <- reboots *now*

halt <- power off the system

reboot <- restart the power system

sync <- writes the data in memory on the filesystem to prevent inconsistencies

- 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).

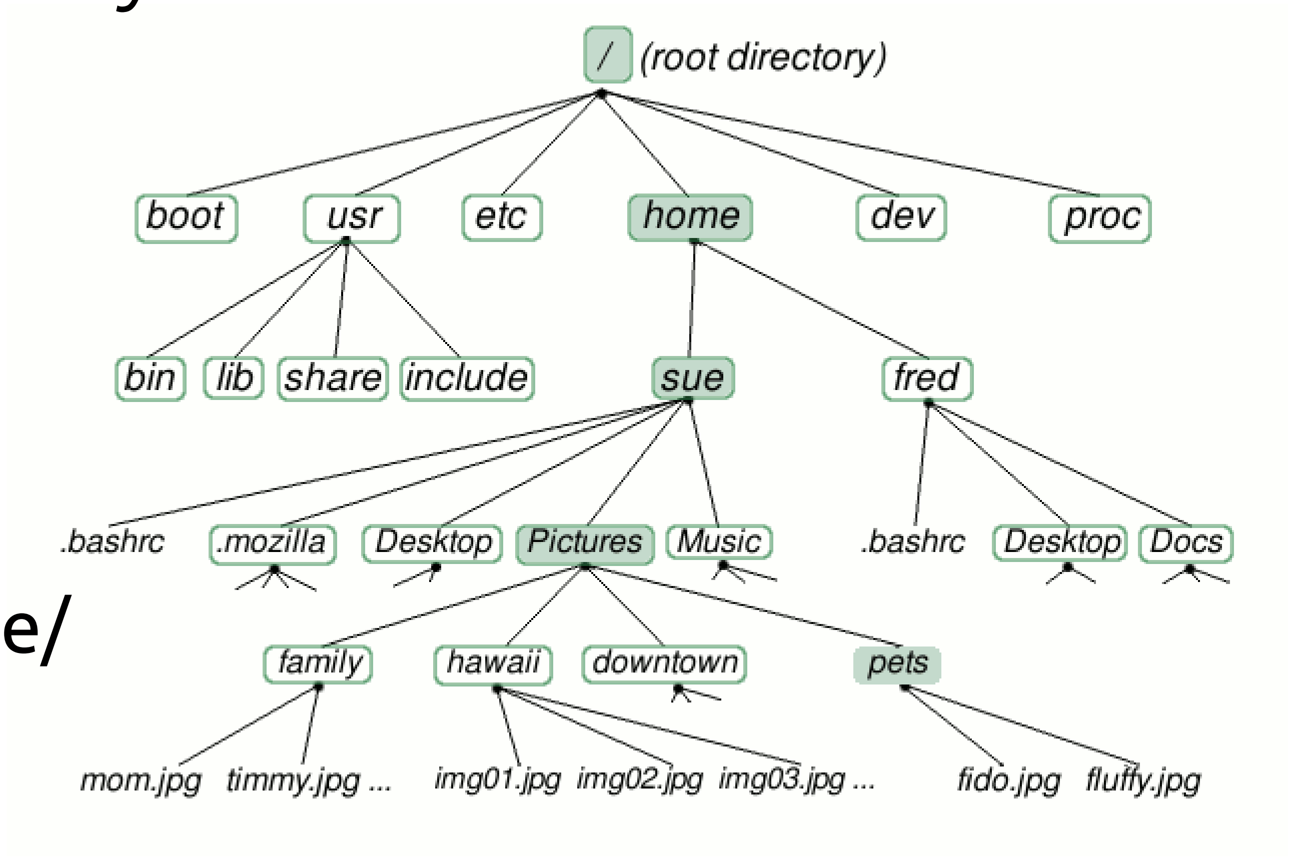
The operation of ***mount*** provide access to **a file** in a certain **position of the tree** (*called the mount point*).

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

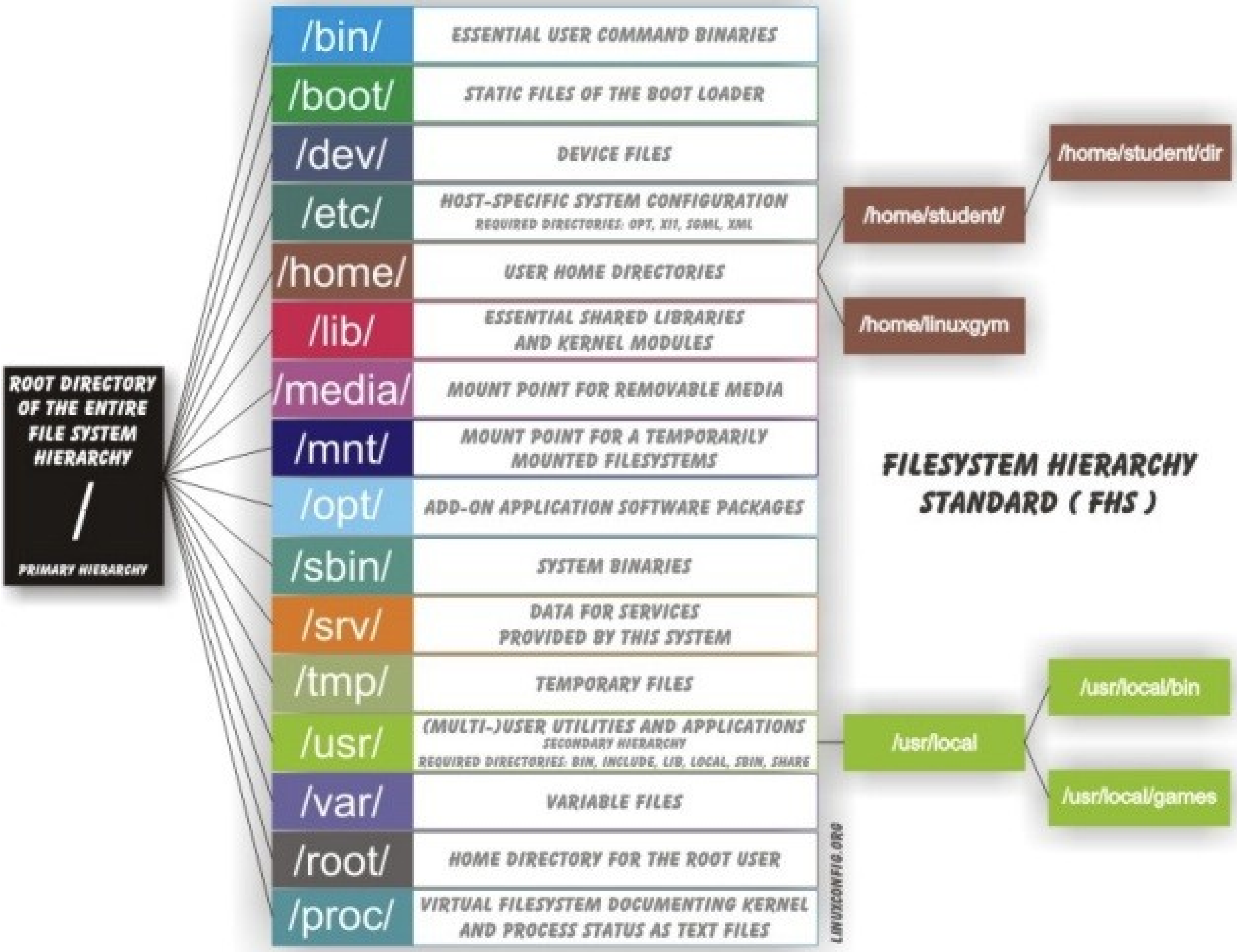
Filesystem structure

- **'/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



ROOT DIRECTORY OF THE ENTIRE FILE SYSTEM HIERARCHY

/

PRIMARY HIERARCHY

/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 HIERARCHY STANDARD (FHS)

LINUXCONFIG.ORG

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.