

```
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
```

```
#define MAXPAROLA 30
#define MAXRIGA 80
```

```
int main(int argc, char *argv[])
```

```
{
    int freq[MAXPAROLA]; /* vettore di contatori
    delle frequenze delle lunghezze delle parole */
    char riga[MAXRIGA];
    int i, inizio, lunghezza;
    FILE *f;
```

```
for(i=0; i<MAXPAROLA; i++)
    freq[i]=0;
```

```
if(argc != 2)
{
    fprintf(stderr, "ERRORE, serve un parametro con il nome del file\n");
    exit(1);
}
```

```
f = fopen(argv[1], "r");
if(f==NULL)
{
    fprintf(stderr, "ERRORE, impossibile aprire il file %s\n", argv[1]);
    exit(1);
}
```

```
while( fgets( riga, MAXRIGA, f ) != NULL )
```



UNIX/Linux Environment

UNIX & Linux commands

Stefano Quer, Pietro Laface, and Stefano Scanzio

Dipartimento di Automatica e Informatica

Politecnico di Torino

skenz.it/os

stefano.scanzio@polito.it

Linux installation

- ❖ Many possibilities exist to setup a UNIX-like (Linux) environment
 - For detailed information search on the WWW the most common Linux versions
 - For example, <https://ubuntu.com/>
 - or the various keywords listed below
- ❖ Main keywords
 - Linux LIVE, multi-boot, Virtual Machine, Windows Linux Subsystem, Docker

Linux installation

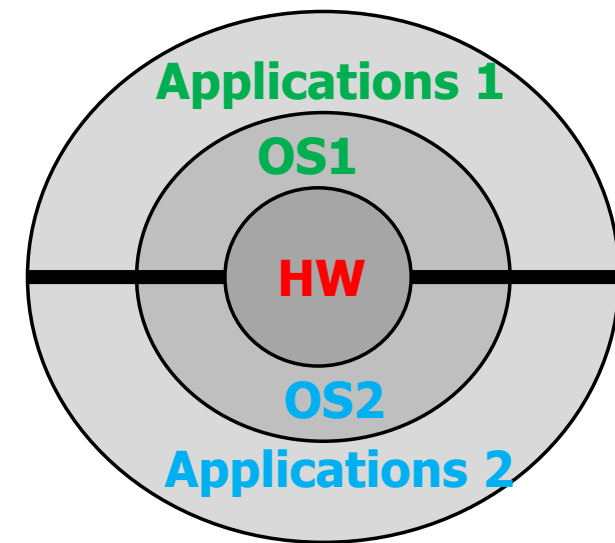
❖ Linux LIVE version

- Practically any modern Linux distribution provides a "LIVE" modality, i.e., the possibility to execute the whole OS without the installation requirement
 - Features are reduced
 - Generally it is not possible to save the system configuration, as every bootstrap is executed from the original state
- In practice Linux is executed from a CD, or (better) a USB-key containing ".iso" and/or other files
- Search "Linux LIVE versions"

Linux installation

❖ Linux in a Multi-boot partition

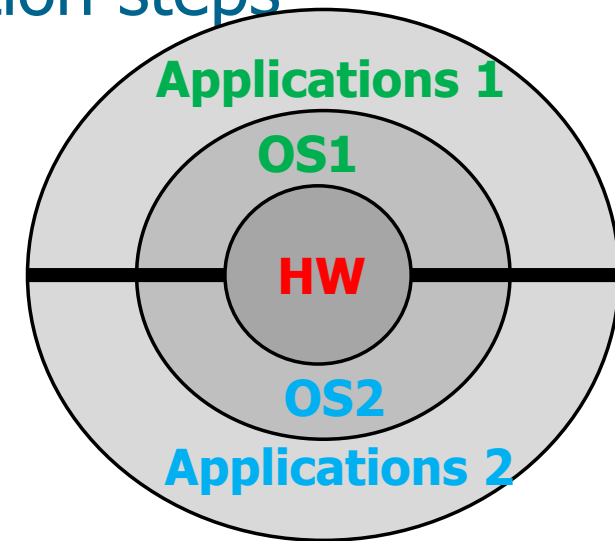
- A disk can be partitioned, and each partition can contain a different OS
 - Complex operation and potentially dangerous
 - During the boot phase, a boot loader
 - GRUB (now **GRUB2**) in GNU Linux
 - NTLDR for Windows NT
- allows to decide with OS to use to bootstrap
- Search "GRUB" or "GRUB2"



Linux installation

❖ Linux in a Multi-boot partition

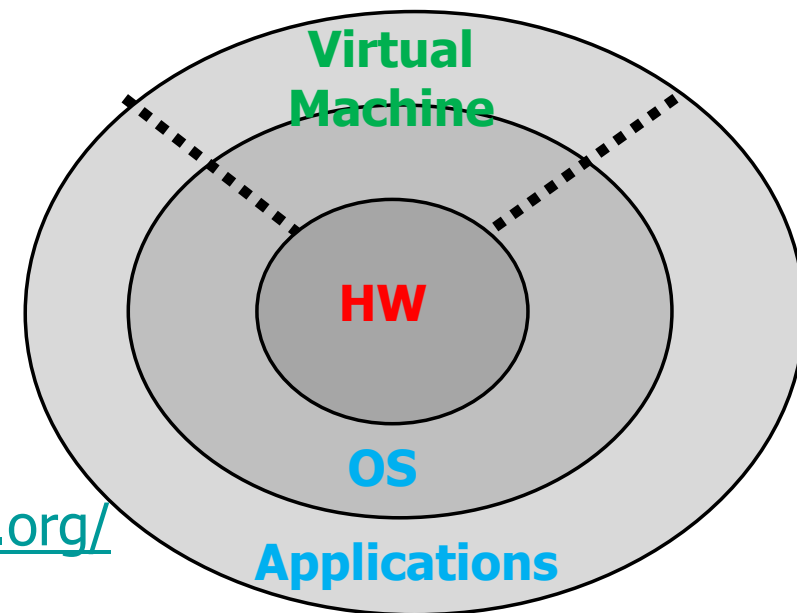
- Practically all "Linux LIVE" version provides the possibility to install permanently the Linux distribution in the harddisk
- In this case the bootloader (i.e., GRUB or GRUB2) is automatically installed in the PC
 - without any need to configure it
- You have just to follow the installation steps



Linux installation

❖ Linux inside a Virtual Machine (VM)

- There are some applications (**virtualizers**) that allow hardware emulation
- The most important
 - Virtualbox (Oracle VM)
 - For AMD64 and Intel64
 - Available for Windows, Linux, MAC, Solaris
 - WWW: <https://www.virtualbox.org/>
 - WMWare (Dell Technologies)
 - Qemu
 - Virtual Machine Microsoft
 - Virsh (CentOS)



Linux installation

- ❖ A virtual machine creates the illusion of the availability of multiple personal computers, each with its own processor and memory
- ❖ Attention
 - Verify that the PC/laptop allows virtualization
 - It can be checked in the bios
 - With VirtualBox, after installing Linux, it is recommended to install the "Guest Addition"
 - Install the related CD and execute it
 - Search: "VirtualBox"

Linux installation

- ❖ Windows Subsystem for Linux (WSL)
- ❖ Originally named “bash on Ubuntu on Windows” or “LXSS, Linux Windows Subsystem”
- ❖ This is not virtualization, because Microsoft has implemented a subsystem which exhibits the same Application Programming Interface (API) of a Linux kernel
- ❖ It is likely more efficient and requires less resources than virtualization
- ❖ Requirements:
 - Windows 10
 - From Windows 10 1607 Anniversary Update (i.e., from 2016)
 - 64 bit version

Linux installation

❖ Windows Subsystem for Linux (WSL)

➤ Installation procedure

- Follow the following or others installation guides
- <https://docs.microsoft.com/it-it/windows/wsl/install-win10>
- <https://ubuntu.com/tutorials/install-ubuntu-on-wsl2-on-windows-11-with-gui-support#1-overview>
- The missing software must be explicitly installed, e.g., for Ubuntu you can use:
 - `sudo apt install <packageName>`

➤ Search: "WLS on Windows 10" or "WLS on Windows 11"

Which linux?

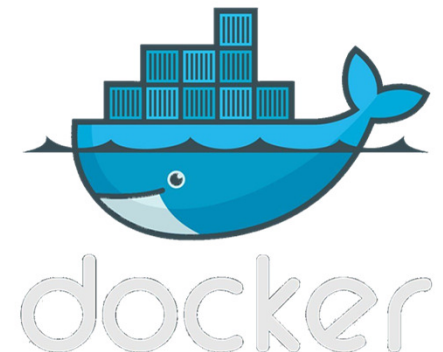
❖ Mint or Ubuntu

- Ubuntu in Nguni Bantu language means "humanity" or "I am because we are"
- A new version released each 6 months
- A new LTS (Long Term Support) version released each 24 months (supported for 5 years)
 - 2014 14.04 LTS Trusty Tahr
 - 2016 16.04 LTS Xenial Xerus
 - 2018 18.04 LTS Bionic Beaver
 - 2020 20.04 LTS Focal Fossa
 - 2022 22.04 LTS Jammy Jellyfish

Updates from April, with different subversion (e.g., 22.04.1, ...)

GNU GPL (General Public Library) license

- ❖ Open-source technology that permits the execution of applications quickly and efficiently.
 - By means of "Containers"
- ❖ Containers
 - Share the kernel of the OS
 - Embed dependencies and configurations of the application



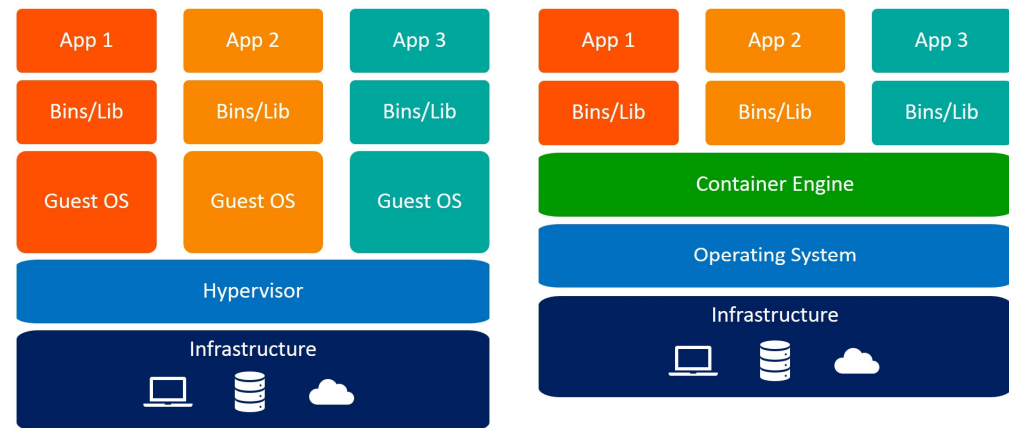
Docker: differences with VM

❖ Docker containers

- Include the application and all its dependences
- But **share the OS**

❖ Virtual machines (VM)

- Include the application, dependences, libraries and the **complete OS**



Virtual Machines

Containers

❖ Docker offers all the advantages of VM

- But requiring fewer requirements from the host OS

Docker: advantages

- ❖ Docker technology offers advantages in terms of:
 - Isolation
 - Portability
 - Agility (less requirements in terms of resources)
 - Scalability
 - Packetization

Docker: an example (Ubuntu)

❖ Execution of Ubuntu

- In Windows, after the installation of Docker, you can run the same commands but without "sudo"
- To install in Linux:
 - `sudo apt install docker`

❖ Run Ubuntu

```
scanzio@light:~$ sudo docker run -it ubuntu bash
root@cee3c7b7e007:/# ls
bin    dev    home  lib32  libx32  mnt    proc  run    srv    tmp    var
boot  etc    lib   lib64  media   opt    root  sbin   sys    usr
root@cee3c7b7e007:/# cd tmp/
root@cee3c7b7e007:/tmp# mkdir test
root@cee3c7b7e007:/tmp# ls
test
root@cee3c7b7e007:/tmp# exit
exit
scanzio@light:~$
```

Docker: an example

❖ List of available containers

```
scanzio@light:~$ sudo docker container ls -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
cee3c7b7e007	ubuntu	"bash"	41 seconds ago	Exited (0) 28 seconds ago		eager_chatterjee

❖ Run an already created container

```
scanzio@light:~$ sudo docker start cee3c7b7e007
cee3c7b7e007
scanzio@light:~$ sudo docker exec -it cee3c7b7e007 bash
root@cee3c7b7e007:/# cd tmp/
root@cee3c7b7e007:/tmp# ls
test
root@cee3c7b7e007:/tmp# exit
exit
scanzio@light:~$ sudo docker container ls -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
cee3c7b7e007	ubuntu	"bash"	2 minutes ago	Up 31 seconds		eager_chatterjee

```
scanzio@light:~$
```

❖ Remove a container

```
scanzio@light:~$ sudo docker container ls -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
cee3c7b7e007	ubuntu	"bash"	2 minutes ago	Up 31 seconds		eager_chatterjee

```
scanzio@light:~$
scanzio@light:~$ sudo docker stop cee3c7b7e007
cee3c7b7e007
scanzio@light:~$ sudo docker rm cee3c7b7e007
cee3c7b7e007
scanzio@light:~$ sudo docker container ls -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

```
scanzio@light:~$
```

Session

❖ Session opening

- login: <username>
- password: <password>

Linux is case sensitive

❖ Remote connection

- ssh <username@hostname> (command line interface; -X option for redirect graphical content)
- ssh -X <username@hostname> (for the redirection of graphical content, Display X11)
- putty (in Windows, graphical interface)

❖ Session termination

- exit
- logout
- ctrl-d

both use a secure encrypted connection protocol

Help manual

❖ All commands are documented in manual pages

- `man <command>`

➤ Related commands

- `apropos <command>`
- `whatis <command>`
- `whereis <command>`

➤ Many commands allow the help option

- `command --help`
and the "version" option
- `command --version`

e.g.,
`man ln`
`man wc`
...

Commands

❖ Unix-like command syntax

`command [options] [arguments]`

❖ The name of the command is associated to the action performed

❖ The options (optional, 0 or more) have conventionally two formats

- The character '-' followed by only another character

- `-ch1 -ch2 ...`

- The two characters "--" followed by a string

- `--str1 --str2 ...`

Or also
`-ch1 ch2 ch3`

❖ Arguments are optional (0 or more)

Commands

❖ Available

- Automatic command completion (Tab)
- Up-down arrows for retrieving previously submitted commands

❖ Command parsing

- Long commands can be continued on the next line using ' \' as the last character of the current line
- Two or more commands can be given on the same line, separated by ' ; '
- `command1 ; command2 ; ...`
- Commands on the same line are executed **sequentially**

Filenames

❖ A filename can include any character sequence

➤ Filenames are case-sensitive

➤ Typically include

- Letters, digits, points '.', underscores '_'

➤ Some characters should not be used

Space / \ " ' * ; ? [] () ~ ! \$ { } < > #
@ & |

➤ The character '\' is reserved as a separator (for directories in paths)

Filenames

❖ Name of the file

- That starts with point "." identifies a hidden, and usually it is not listed
- Has a limited length (usually 255 characters)
- Must be unique within the directory

❖ Obsolete files (for example those created by autosave) are often automatically renamed by postponing the character ~ to the name

Filenames

- Formally a file has not extension and version
- Some meaningful extension are often used
 - `.c`, `.cpp`, `.sh`, `.o`, `.a`, `.so`, `.awk`, `.tar`,
`.gz`, `.tgz`, `a.out`, `core`
- ❖ A filename beginning by ' .' corresponds to an **hidden** file, i.e., a file that is not normally visible listing the content of a directory

Filesystem

❖ The Linux filesystem is

- Hierarchic
- Organized by means of tree directories
 - The root tree directory is ' / ' (slash)
 - The current directory is indicated by ' . ' (dot)
 - The parent directory is indicated by ' . . ' (dot dot)
 - Directories are separated by means of a ' / ' (slash)
- Uniform notation for disks, directories, files, special files, ...

❖ A file is specified by its pathname

➤ Absolute pathname

- From the filesystem root
- `/dir1/dir2/file`

➤ Relative pathname

- From the current working directory
- `./subdir1/subdir2/file`
- `subdir1/subdir2/file`

Regular file management: ls

- ❖ Command **ls** provides information about a file according to the specified options. If pathname is a directory, **ls** lists the files and subdirectories contained in that directory (i.e., the 'entries' of a directory)

ls [-options] [file ...]

➤ Options

- --help
 - in-line help
- --all, -a
 - Shows also hidden files (filenames beginning with '.')
- -l
 - Long list format (extended output)

Regular file management: ls

- `--group-directories-first, -g`
 - Included group info before those related to files
- `-t`
 - Sort files by date (newest first)
- `--reverse, -r`
 - Reverse order (alphabetic/date)
- `--recursive, -R`
 - Recursive (includes files in subdirectories)

Example

List of type
"long-list-format"
for "all-files"

```
$ ls -la
total 72
drwxr-xr-x  8 user1 group1 4096 Oct  7 2013 .
drwxr-xr-x 34 user1 group1 4096 Oct  3 12:37 ..
drwxr-xr-x  2 user1 group1 4096 Oct 15 2009 file
-rw-r--r--  1 user1 group1 17715 Oct  7 2013 index.htm
drwxr-xr-x  2 user1 group1 4096 Mar 22 2013 misc
drwxr-xr-x  2 user1 group1 4096 Jun 25 2009 paper
drwxr-xr-x  3 user1 group1 4096 May 30 2012 research
-rw-r--r--  1 user1 group1 18074 Apr 28 2005 stq.jpg
drwxr-xr-x 10 user1 group1 4096 Jun  5 14:56 teaching
drwxr-xr-x  2 user1 group1 4096 Jun  2 20:49 tmp
```

The "ls" command would
provide only the list of
files in the directory

Example

Total Number of Blocks
(default size 1024 bytes)

User (owner)
name

Owner
group

Entry name

```
$ ls -la
total 72
drwxr-xr-x  8 user1 group1 4096 Oct  7  2013 .
drwxr-xr-x 34 user1 group1 4096 Oct  3 12:37 ..
drwxr-xr-x  2 user1 group1 4096 Oct 15  2009 file
-rw-r--r--  1 user1 group1 17715 Oct  7  2013 index.htm
drwxr-xr-x  2 user1 group1 4096 Mar 22  2013 misc
drwxr-xr-x  2 user1 group1 4096 Jun 25  2009 paper
drwxr-xr-x  3 user1 group1 4096 May 30  2012 research
-rw-r--r--  1 user1 group1 18074 Apr 28  2005 stq.jpg
drwxr-xr-x 10 user1 group1 4096 Jun  5 14:56 teaching
drwxr-xr-x  2 user1 group1 4096 Jun  2 20:49 tmp
```

Type &
permissions

Number of links

Size (in byte)

Last modification
date

Example

File type

-	Normal file
d	Directory
s	Socket file
l	Link file

Three users types

u	user	(owner)
g	group	
o	others	other users

```
...  
-rw-r--r-- 1 user1 group1 17715 Oct 7 2013 index.htm  
drwxr-xr-x 2 user1 group1 4096 Mar 22 2013 misc  
...
```

Three base permissions

r	read
w	write
x	execute

Example

Permission can be defined as an octal value

rwX rwX rwX	→	777
rw- rw- rw-	→	666
rwX --X ---	→	710

```
...  
-rw-r--r--  1 user1 group 17715 Oct  7  2013 index.htm  
drwxr-xr-x  2 user1 group 14096 Mar 22  2013 misc  
...
```

Alternatively by means of

- a letter: u(ser), g(roup), o(ther), a(ll)
- a symbol: +, -, = (add, subtract, untouched)
- a character: r, w, x (read, write, execute)

(see chmod command)

Regular file management

❖ Copy a file

- `cp [options] src1 src2 ... dest`
- Example
 - `cp file1 file2 file3 ... dir`

❖ Remove a file

- `rm [options] file1 file2 ...`

❖ Move (rename) a file

- `mv [options] file1 file2 ... dest`

Regular file management

❖ Options

- --help
 - in-line help
- --force, -f
 - does not ask confirmation (force)
- --interactive, -i
 - ask confirmation for each file (interactive)
- --recursive, -r, -R
 - Apply command recursively on all the subdirectory files

rm over objects without write rights requires confirmation

❖ Directories can often be managed as regular files

Regular file management

❖ Examples

```
cp pippo topolino  
cp -R dir1 dir2
```

```
rm pluto.txt  
rm -rf dir1 dir2 dir3
```

```
mv paperino.c paperina.c
```

rm applied to objects
without write permission
requires confirmation

Directory management

- ❖ Change current directory
 - `cd dir`
- ❖ Print working directory
 - `pwd`
- ❖ Create a directory
 - `mkdir dir`
- ❖ Remove a directory
 - `rmdir dir`
 - A directory can be removed only if it is empty, unless the options `-fr` are used with command
 - `rm -fr dir`

Permissions for directories

❖ The meaning of the permission "rwx" is different between files and directories

➤ File

- r
 - Read permission (of the file)
- W
 - Write permission (of the content of the file)
- X
 - Execution permission (the file can be executed)

cp file1 file2
fails if file1 has not read
permissions or if file2 has not
write permissions

Permissions for directories

❖ The meaning of the permission characters is different for directories.

➤ Directory

- **r**
 - Directory content can be listed
- **w**
 - Create, rename, or delete files within the directory
- **x**
 - Directory can be crossed or **cd** command can be performed (to access it, not to list)

cd dir
fails if dir has not
execution permissions

Permission management

- ❖ It is possible to change file permissions if you have the rights, i.e., if you are the owner of the file
- ❖ There are commands to change personal generalities (i.e., the user) of files on a UNIX system
 - To become a different user
 - `su username`
 - The password of the new user is requested
 - To run commands as a superuser (or other user)
 - `sudo command`
 - The password of the user is required
 - To know which user you are
 - `whoami`

The super-user do the command:
`sudo -u user command`

Permission management

- ❖ It is possible to change file permissions
 - `chmod [options] permissions file`
- ❖ Permissions can be specified in different ways
 - Absolute, by means of three octal digits
 - `chmod 775 filename`
 - Symbolic, by means of a string of three (or more) characters
 - `chmod g+r filename`
 - `chmod +x filename`
 - `chmod +xw filename`
 - `chmod uo+rx filename`

u (user)		r		+
g (group)	→	w	→	-
o (other)		x		=
a (all)				

uo+rx: Add (+) to user (u) and other (o)
the read (r) and execute (x) permissions

Permission management

❖ Examples

```
chmod g+r nomefile  
chmod +x nomefile  
chmod +xw nomefile  
chmod u+rx nomefile
```

For the group add the
read permission

For the user and other add
the read and execution
permission

Permission management

- ❖ Changing the owner of a directory entry
 - `chown [options] user entry`
- ❖ Changing the group of a directory entry
 - `chgrp [options] group entry`
- ❖ These command can be combined
 - `chown [options] user[:group] entry`
 - `chown [options] uid[:gid] entry`
- Options
 - `--recursive, -R`
 - Performed recursively on all entries of the directory tree

Output the content of a file

❖ Output and concatenate files

- `cat filename1 filename2 ...`

❖ Output the first **num** lines of a file

- `head [options] filename ...`

Default num=10

❖ Output the last **num** lines of a file

- `tail [options] filename ...`

`tail -n 2 file`
prints the last two lines of file
Compact version: `tail -2 file`

➤ Options

- `--lines num, -n num`
 - print first (head) / last (tail) num lines
- `--follow, -f`
 - outputs appended data as the file grows (i.e., the file is continuously re-read)

Output the content of a file

❖ Additional output commands

- `pg [options] filename ...`
 - "browse page-wise through text file"
- `more [options] filename ...`
 - to view a text file
- `less [options] filename ...`
 - Like the previous command but allows the use of arrows to move in the text (advanced version of more)

Output the content of a file

- Some commands when a file is opened with less or more commands
 - space Next page
 - return Next line
 - b Previous page
 - /str Find next occurrence of string str
 - ?str Find previous occurrence of string str
 - q Quit

File comparison

❖ Difference between two files

- `diff [options] file1 file2`

➤ Lists the line number of the lines

- a
 - added
- d
 - deleted
- c
 - changed

❖ Difference between two directories

- `diff [options] dir1 dir2`

File comparison

➤ Options

- --brief, -q
 - Reports only when files differ (default)
- --ignore-space-change, -b
 - Ignores spaces at the end of the line, merges the others
- --ignore-case, -i
 - Case insensitive
- --ignore-all-space, -w
 - Ignores completely all white spaces
- --ignore-blank-lines, -B
 - Ignores all blank lines

Counts

❖ Outputs the number of lines, words, and bytes of a file

- `wc [options] [file...]`

➤ Options

- `--lines, -l`
 - Outputs only the number of lines
- `--words, -w`
 - Outputs only the number of words
- `--bytes, -c`
 - Outputs only the number of bytes
- `--chars, -m`
 - Outputs only the number of characters
 - Option typically not used

Warning: it also outputs the filename as its first line

Hard and Soft Link

❖ There are two types of links in UNIX

➤ Symbolic or soft link

- Particular type of file that simply contains a path (i.e., the name) of another object (file or directory)
- Allows references between different file-systems (partitions)
- If you remove the file the link remains pending

➤ Physical or hard link

- Association between an object name and its content (pointer from directory-entry to i-node)
- It is not possible to create hard links between different file-systems, or hard links to a directory
- The file is removed only when it is removed the last of its hard links

Hard and Symbolic Links

❖ Link creation

- In [options] source [destination]

➤ Default behavior

- Creates a hard link
- If the destination is not present, creates a link with the same filename on the working directory

Hard and Symbolic Links

➤ Options

- --help
 - in-line help
- --symbolic, -s
 - Creates a symbolic link (soft link)
- --force, -f
 - Force creation, removes file if already exist
- --directory, -f, -F
 - allow the superuser to attempt to create a hard link to a directories (note: will probably fail due to system restrictions, even for the superuser)

Hard and Symbolic Links

➤ Examples

- `ln source alias`
- `ln /home/scanzio/file`
 - Corresponds to `ln /home/scanzio/file .`
- `ln -s /home/foo/tmp/bar.exe /mnt/foo/bin/`

Symbolic link, possibly to a file in another filesystem

❖ Notice that

➤ Command **rm**

- Removes the data of a file only if its link number is equal to 0

➤ Command **mv**

- Performed as the sequence of commands **ln** followed by **rm**

Hard and Soft Link

❖ Observe that

➤ The command **rm**

- Removes a file only if the number of hard link is equal to 0

➤ The command **mv**

- Is equivalent to execute firstly the command **ln** and then the command **rm**

Archive management

- ❖ Data storage and compression can be managed using the **tar** command
 - tar = an archiving utility

Archive management

- Archiving and compression of the files in the directory `dir`, in a file with name `file.tar.gz`
 - `tar czvf <file>.tar.gz <dir>`
- Useful options
 - `c`
 - Creates the archive
 - `z, j, J`
 - Compression (gzip, bzip2, 7z)
 - 7z allows to reach really high compression rates
 - `v`
 - Verbose (print some messages and statistics)
 - `f`
 - Specify the name of the archive (always present)

Archive management

➤ Extract the content of an archive

- `tar xzvf <file>.tgz <dir>`

➤ Useful options

- `x`
 - Extracts the files from the archive
- `z, j, J`
 - Compression (gzip, bzip2, 7z)
- `v`
 - Verbose (print some messages and statistics)
- `f`
 - Specify the name of the archive (always present)

Archive management

❖ Alternative commands

- gzip, gunzip
- zip, unzip
- rar, unrar
- compress

Disk space occupation

❖ To control disk occupancy, it is possible to use the **df** command

- `df [options] [disk ...]`

File system
disk space
usage

➤ Options

- `--block-size=SIZE, -B SIZE`
 - scale sizes by SIZE before printing them. SIZE can be, e.g., 1K, 10K, 1M, 1G, 1T, etc.
- `-k`
 - corresponds to `--block-size=1K`

Example

```
$ df
```

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
udev	8183252	0	8183252	0%	/dev
tmpfs	1642600	9248	1633352	1%	/run
/dev/sda1	49808620	14095784	33159648	30%	/
tmpfs	8212992	220	8212772	1%	/dev/shm
tmpfs	5120	4	5116	1%	/run/lock
tmpfs	8212992	0	8212992	0%	/sys/fs/
F_DRIVE	600948732	260043768	340904964	44%	/media/D
G_DRIVE	976760828	897641752	79119076	92%	/media/G
tmpfs	1642600	44	1642556	1%	/run/user/

Disk space occupation

❖ To get the space occupied by a directory and all its subdirectories it is possible to use the command

- `du [options] directory ...`

➤ Options

- `--all, -a`
 - Occupation of each file
- `--summarize, -s`
 - Prints only the total
- `--block-size=1K, -k`
 - Occupation in kB

Example

Space
occupied
by files

```
$ du
4      ./run.sh
8      ./wiFiStat.c
4      ./Makefile
4      ./run2.sh
4      ./README
4      ./adhoc.sh
4      ./TAGS
16     ./last_stat.c
4      ./elab_out.py
8      ./main.c
4      ./inc/net.h
...
184    .
```