

Corso di Laurea in Informatica A.A. 2023-2024

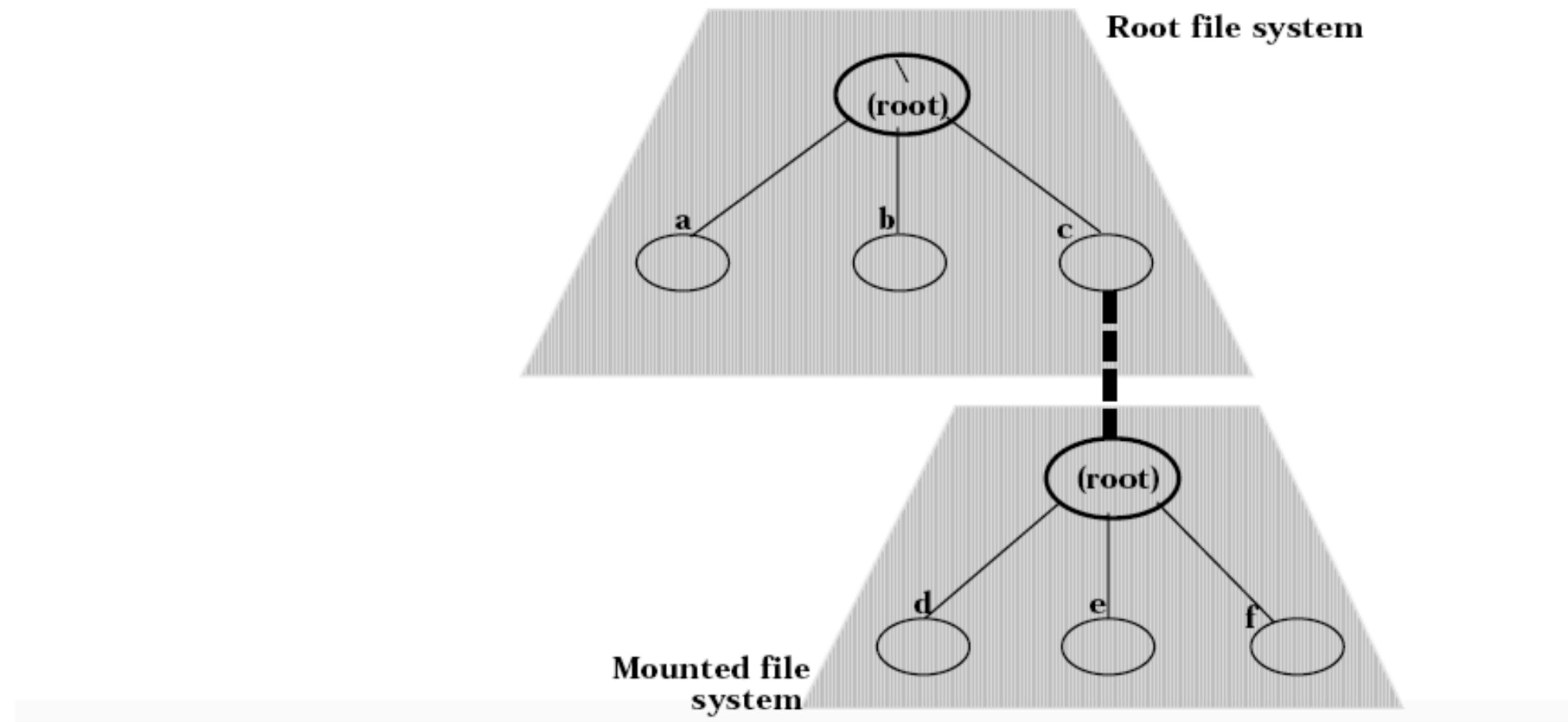
Laboratorio di Sistemi Operativi

Alessandra Rossi

I File System Montabile

Un file system Unix è sempre **unico**, ma può avere parti residenti su device rimuovibili:

- "montate" prima di potervi accedere (comando mount)
- "smontate" prima di rimuovere il supporto (comando umount)



Gestione delle Directory

- **pwd** print working directory
- **cd** change directory
- **ls** list directory
- **du** disk usage
- **mkdir** make directory
- **rmdir** remove directory
- **ln** link

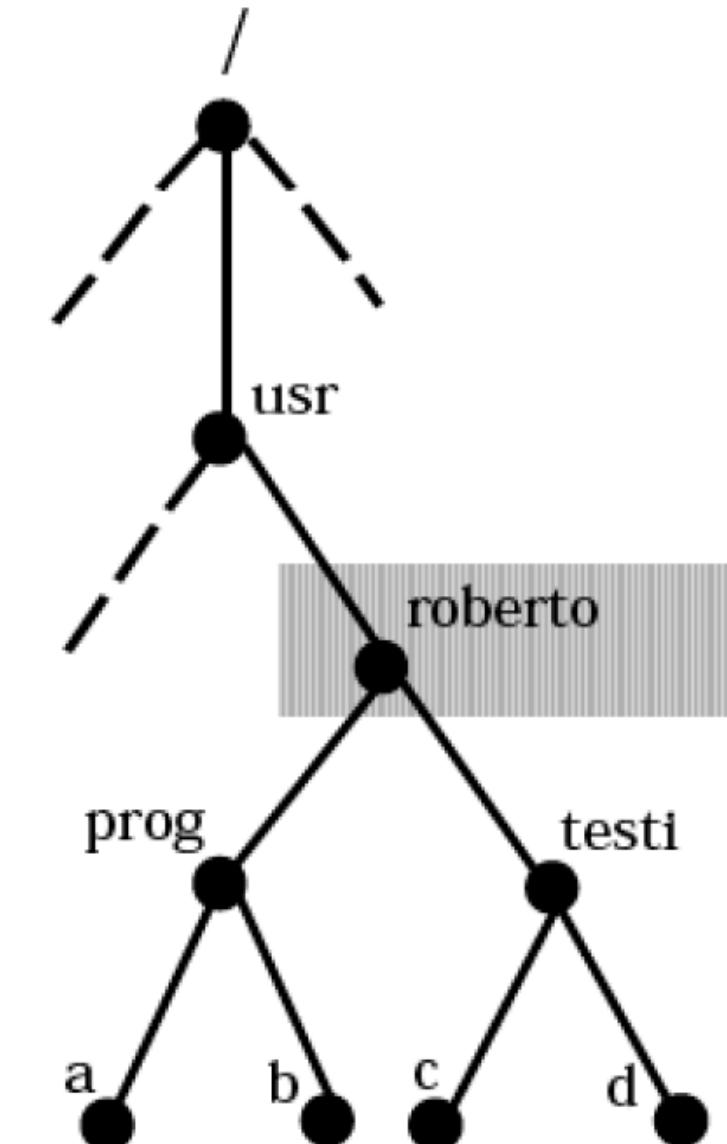
PWD

pwd (print working directory)

- Stampa pathname directory corrente

Esempio:

```
% pwd  
/usr/roberto  
%
```



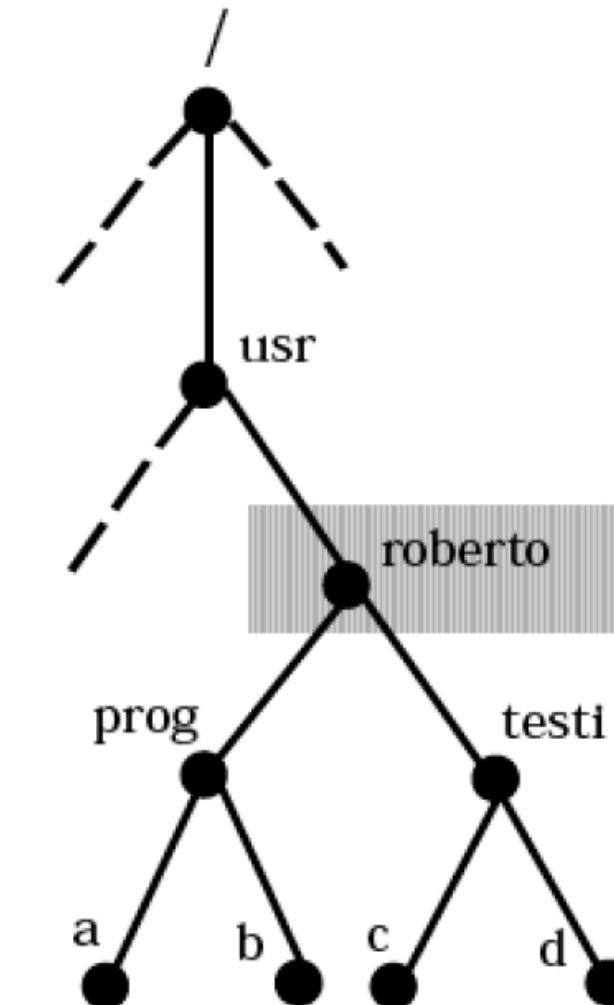
CD

cd change directory

- La directory specificata diviene la working directory
- se nessuna directory specificata, si "ritorna" alla home directory

Esempio:

```
%cd /usr  
%pwd  
/usr  
%cd  
%pwd  
/usr/roberto  
%
```



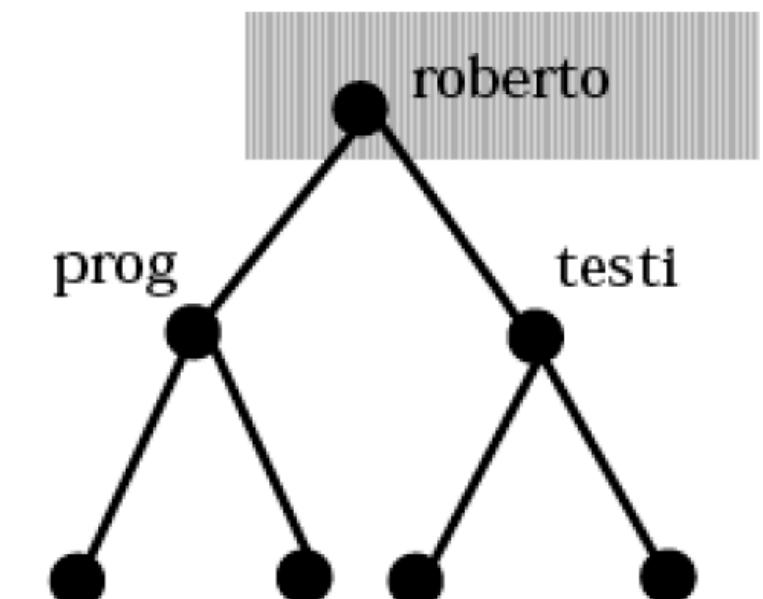
LS

ls [options] [directory1 file2 ...]

- Lista (in ordine alfabetico) il contenuto della o delle directories indicate
- Accetta anche nomi di file
- Senza parametri, elenca il contenuto della working directory
- Possiede numerose opzioni

Esempio:

```
% ls  
prog testi  
%
```



I Comandi UNIX

- **-l** (**long**) formato esteso con informazioni aggiuntive
 - **-a** (**all**) mostra anche i file “nascosti” (dotfiles)
-
- **-R** (**Recursive**) visita ricorsivamente le sottodirectory
 - **-i** mostra l'**i-number**
 - **-t** (**time**) lista nell'ordine di modifica (prima il file modificato per ultimo)
... e molte altre

Esempio LS

```
% ls
dir1 file1
% ls -s
total 4 2 dir1 2 file1
% ls -t
file1 dir1
% ls -1
dir1
file1
% ls -F
dir1/ file1
% ls -R
dir1 file1
./dir1:
file1 file2 file3 file4
% ls -i
199742 dir1 51204 file1
```

I file nascosti

- I files il cui nome inizia con "." vengono listati
- solo specificando l'opzione -a ("all")
- **Esempio:**

```
% ls -a  
.. .cshrc .mailrc dir1  
... .login .sh_history file1  
%
```

LS esteso

Totale dimensione occupata (in blocchi)

		Riferimenti al file	Dimensione (byte)	Nome
lso:~>ls -l				
total 12				
-rw-rw-r--	1 lso	lso	10 Mar 4 13:29 a	
-rw-rw-r--	1 lso	lso	10 Mar 4 14:12 b	
drwxrwxr-x	2 lso	lso	4096 Mar 4 14:29 c	

Diagramma di espansione dei campi del comando ls -l:

- Tipo:** Indica il tipo di file: (d)irectory, (l)ink, (c)haracter special file, (b)lock special file, (-) ordinary file.
- Permessi:** Indica i permessi di lettura (r), scrittura (w) e esecuzione (x). Ad esempio, "drwxrwxr-x" indica un directory con permessi per tutti.
- Riferimenti al file:** Indica il numero di riferimenti al file.
- Proprietario:** Indica il proprietario del file.
- Gruppo:** Indica il gruppo del file.
- Dimensione (byte):** Indica la dimensione del file in byte.
- Data ultima modifica:** Indica la data e l'ora della ultima modifica del file.
- Nome:** Indica il nome del file.

(d)irectory, (l)ink, (c)haracter special file, (b)lock special file, (-) ordinary file

Protezione di un File

A ciascun file (normale, speciale, directory) sono associati alcuni attributi:

- **Proprietario (owner)**: l'utente che ha creato il file
- **Gruppo (group)**: il gruppo a cui il proprietario appartiene
- **Permessi (permissions)** Il tipo di operazioni che il proprietario, i membri del suo gruppo o gli altri utenti possono compiere sul file

Proprietario, gruppo e permessi iniziali sono assegnati dal sistema al file al momento della sua creazione.

Il proprietario può successivamente modificare tali attributi con appositi comandi (**chown, chgrp, chmod**)

Identificazione Utenti

- Username
- Password
- ID
- Gruppo

Identificazione Utenti

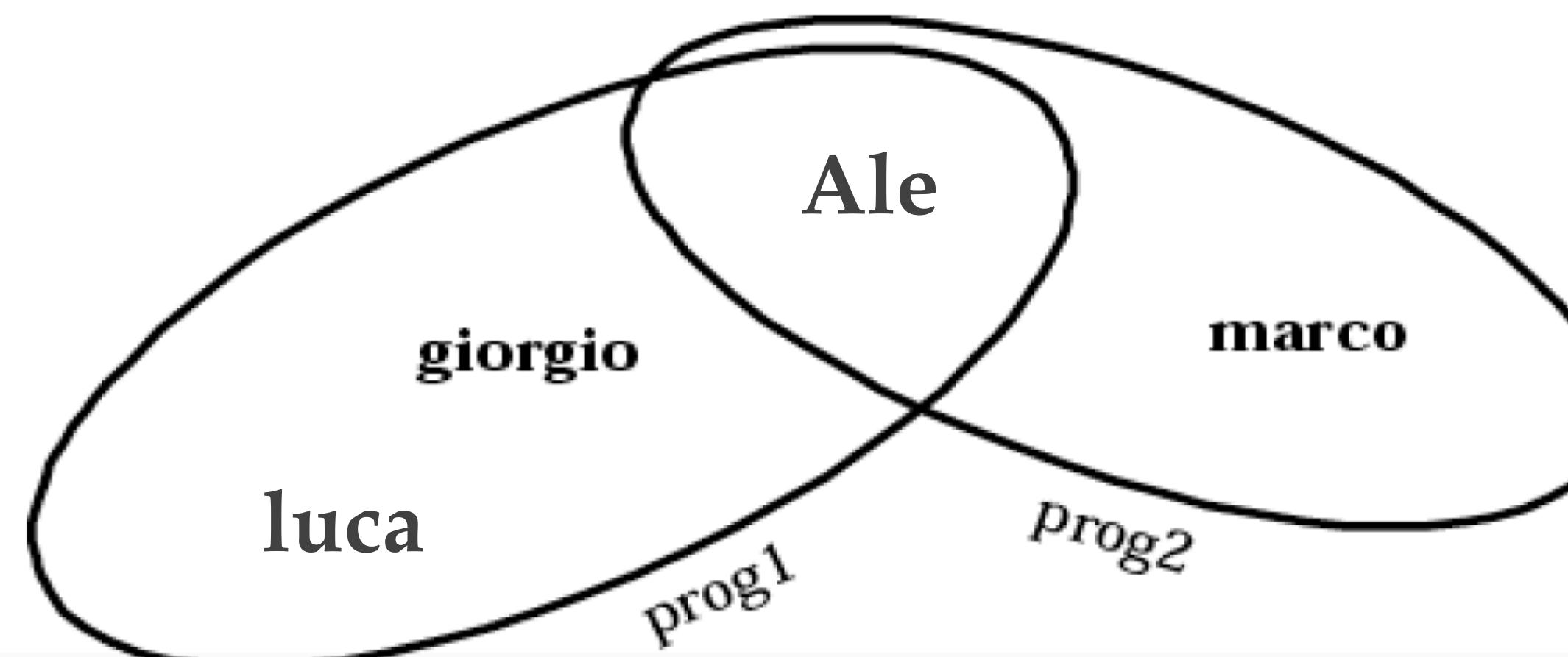
- Ogni utente viene identificato da uno **user name** assegnato dall'amministratore del sistema. Ad esso corrisponde biunivocamente uno **userid** numerico, assegnato dal sistema
- User name e user-id sono **pubblici**

I Gruppi

Ogni utente può far parte di uno o più **gruppi**, definiti dall'amministratore del sistema

Ogni gruppo è identificato da un **group name** di al più 8 caratteri, associato biunivocamente a un **group-id** numerico

Esempio:

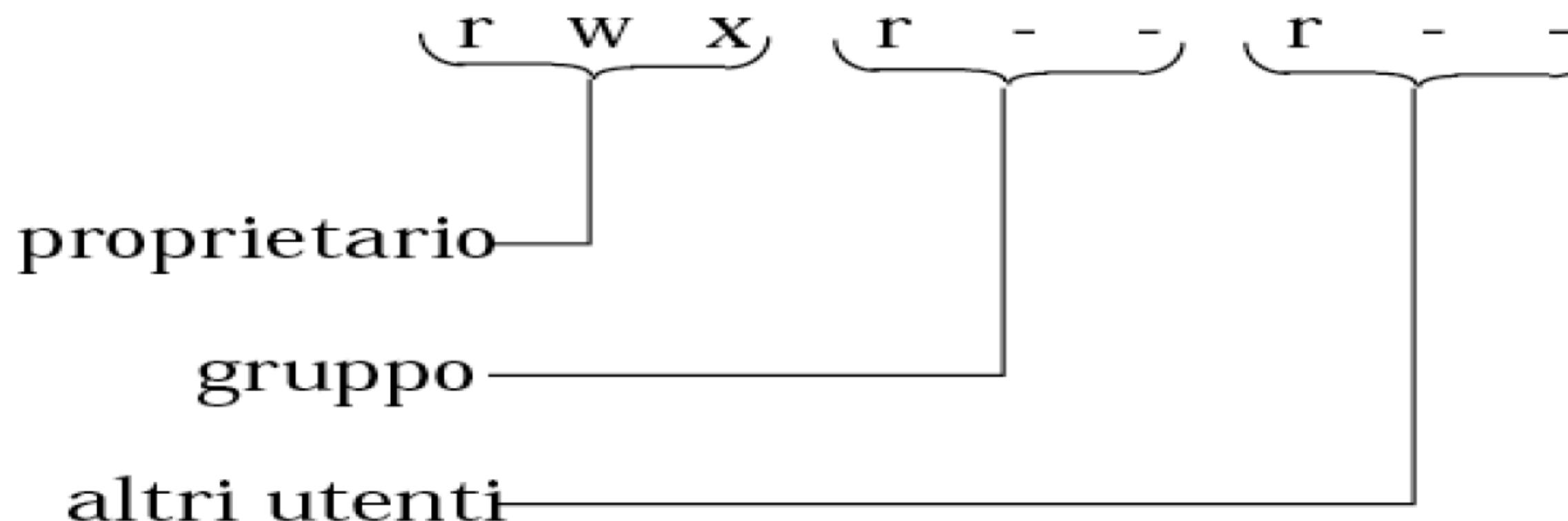


I Permessi

Ad un file possono essere attribuiti i seguenti permessi:

r : readable	per	proprietario gruppo altri utenti
w : writable		
x : executable		

Esempio:



In binario: 1 1 1 1 0 0 1 0 0

In ottale: 7 4 4

I Permessi

- Alla creazione di un file, Unix assegna i seguenti permessi:
- Per i *files ordinari non eseguibili*:

rw-rw-rw

110 110 110

6 6 6

- Per i *files ordinari eseguibili* e per directories:

rwx rwx rwx

111 111 111

7 7 7

Chmod

chmod *permissions filename...*

"change mode"

- attribuisce le *permissions* a *filename*
(solo da parte del proprietario del file!)
- *permissions* può essere espresso in
forma ottale o simbolica

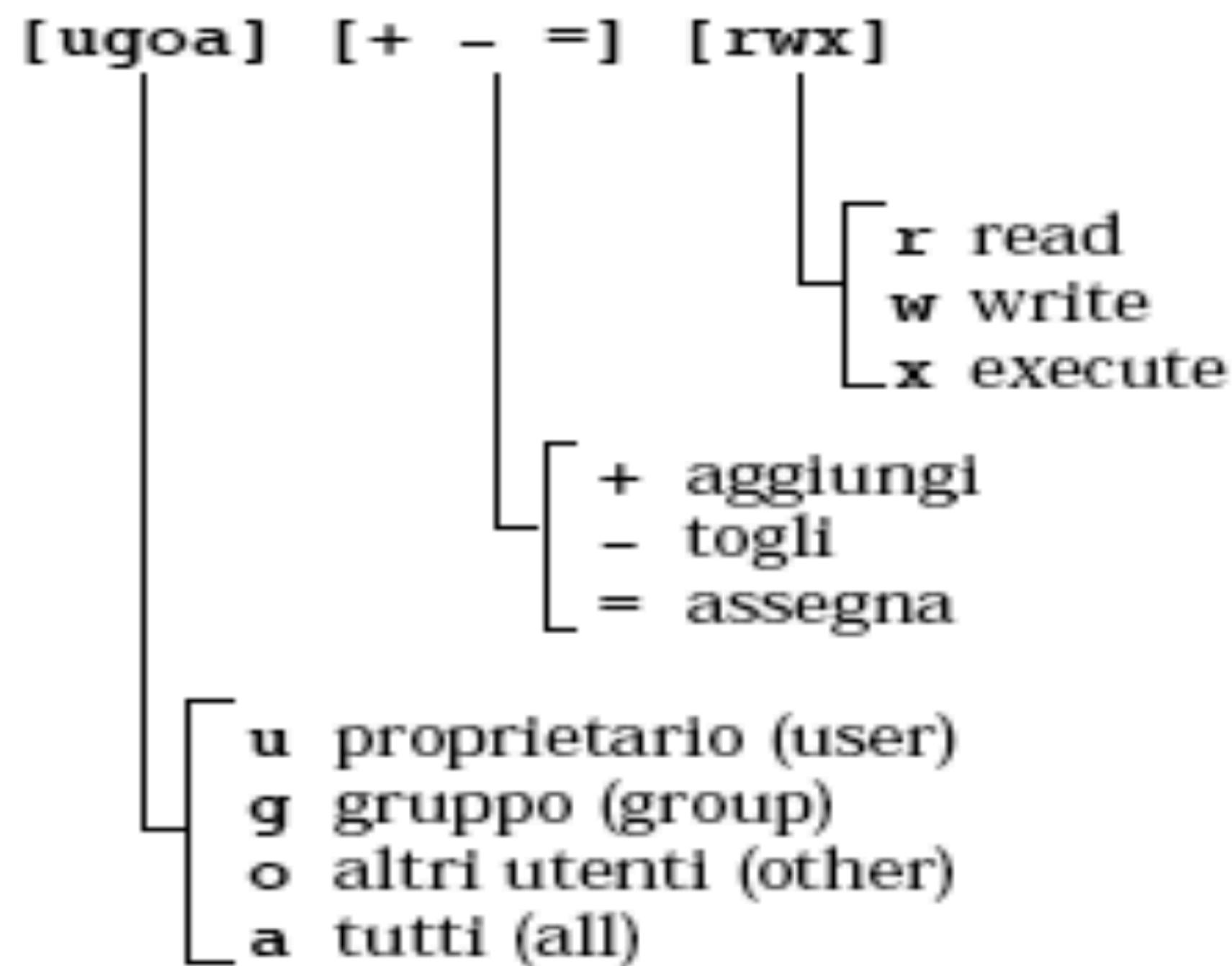
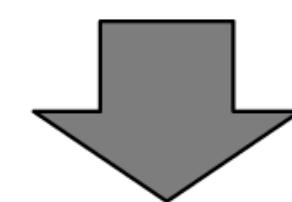
Chmod

chmod <permissions> <filename1 ...>

- Modifica i permessi dei file elencati
- Può farlo solo il proprietario (o root)
- *permissions* può essere espresso in forma ottale o **simbolica**

Chmod

chmod <permissions> <filename1 ...>



Chown

chown (change owner)

```
chown [options] [user] [:group] ]  
file...
```

Cambia proprietario e/o gruppo primario per uno o più file.

Se dopo “:” non segue il nome del gruppo, viene attribuito il gruppo principale cui appartiene user.

Se prima di :group non viene indicato il nome dell'utente, viene cambiato solo il gruppo primario (chgrp)

Chgrp

chgrp *newgroupid file...*

"change group"

- *newgroupid* diventa il nuovo gruppo dei *file...*
- il comando può essere eseguito solo dal proprietario (o dal superuser)

Mk e Rm dir

mkdir e rmdir

- **mkdir** directory ... : Crea la/le directory
Esempio:

```
% mkdir dir1 dir2
% ls
dir1  dir2
```

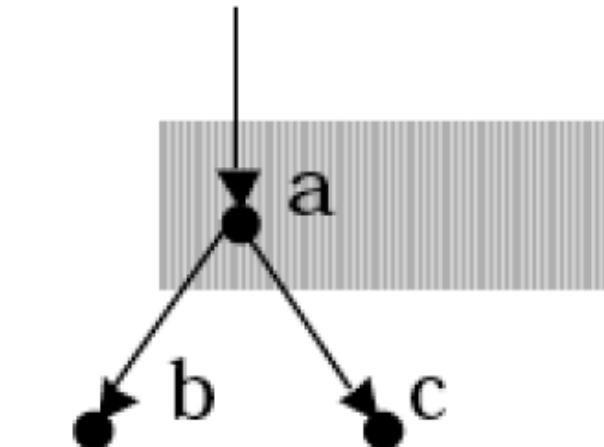
- **rmdir** directory ... : rimuove la/le directory
(deve essere vuota)

```
% rmdir dir
rmdir: dir: Directory not empty
% ls dir
a
% rm dir/a
% rmdir dir
```

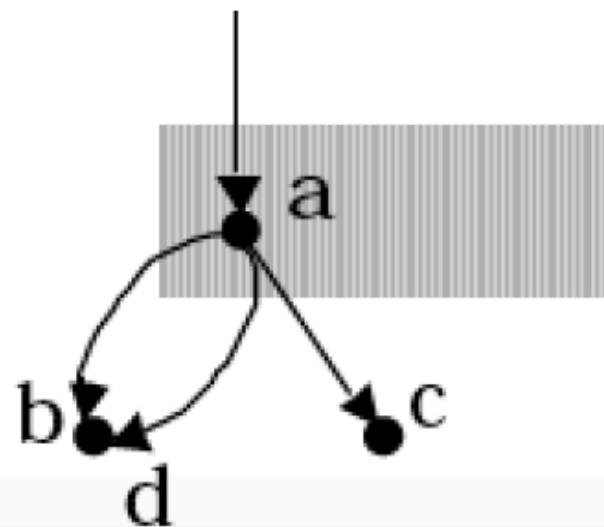
LN

ln name1 name2

associa il nuovo nome (link) name2 al file (esistente) name1, che non può essere una directory



```
% ln b d  
%
```



LN

- Tutti i link allo stesso file hanno identico status e caratteristiche
- Non è possibile distinguere la entry originaria dai nuovi link
- I link di questo tipo non possono essere fatti con file che stanno su file system diversi

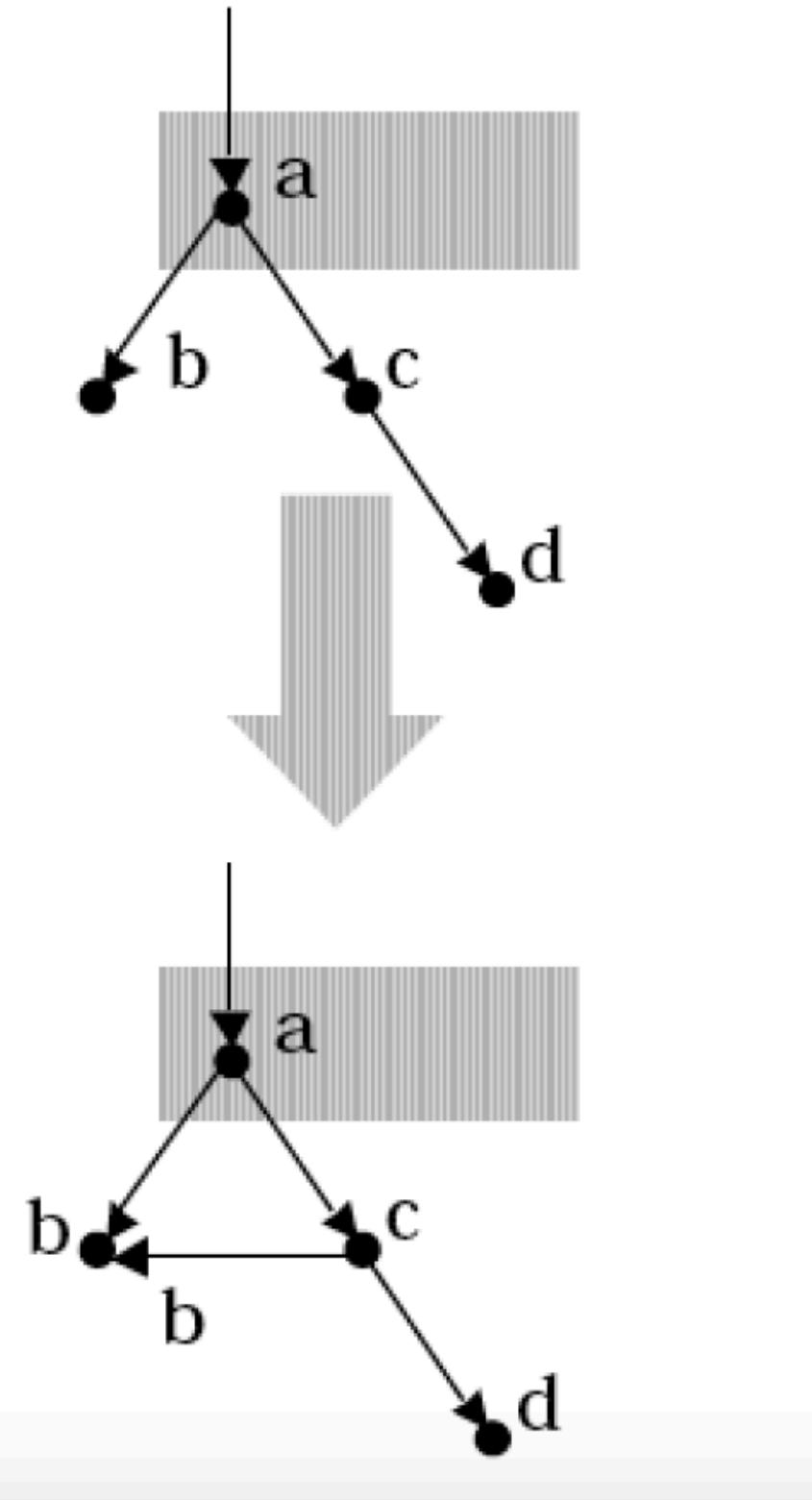
LN

ln name1 name2

Se name2 è una directory, il nuovo nome è
name2/name1

"link"

```
% ln b c  
%
```

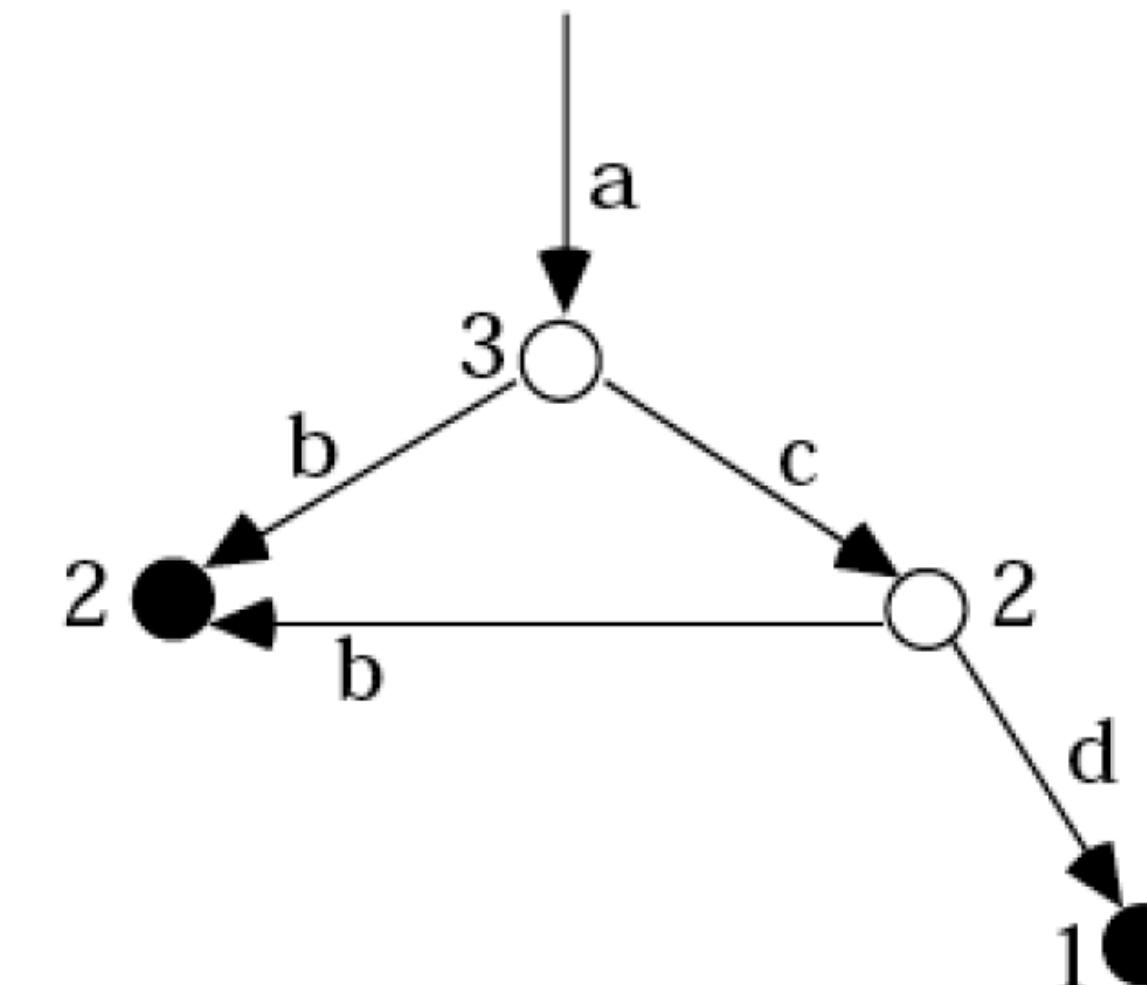


Ln

- Numero links e' un attributo gestito dal sistema

Per vedere:

ls -l



○ directory
● file

I Comandi UNIX

```
% mkdir dir  
% touch file  
% ls -l  
total 2  
drwxr-sr-x  2 roberto  usrmail    512  Mar 11 19:40 dir  
-rw-r--r--  1 roberto  usrmail     0  Mar 11 19:40  
file
```

```
% ln file nuovo
```

Crea link a file da nuovo

```
% ls -i
```

```
199742 dir  51204 file  51204 nuovo
```

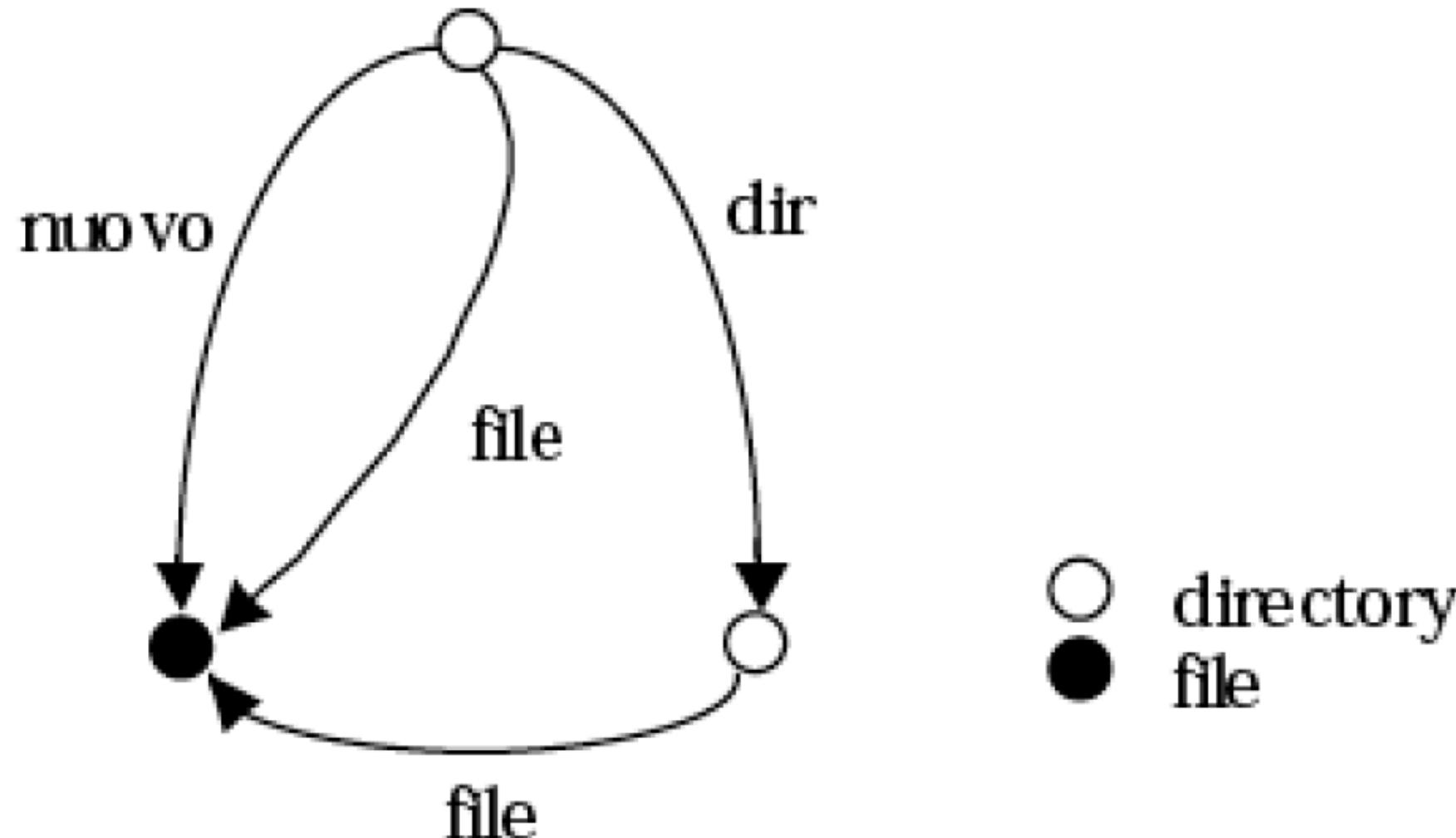
```
% ls -l
```

```
total 2
```

```
drwxr-sr-x  2 roberto  usrmail    512 Mar 11 19:40 dir  
-rw-r--r--  2 roberto  usrmail     0 Mar 11 19:40 file  
-rw-r--r--  2 roberto  usrmail     0 Mar 11 19:40  
nuovo
```

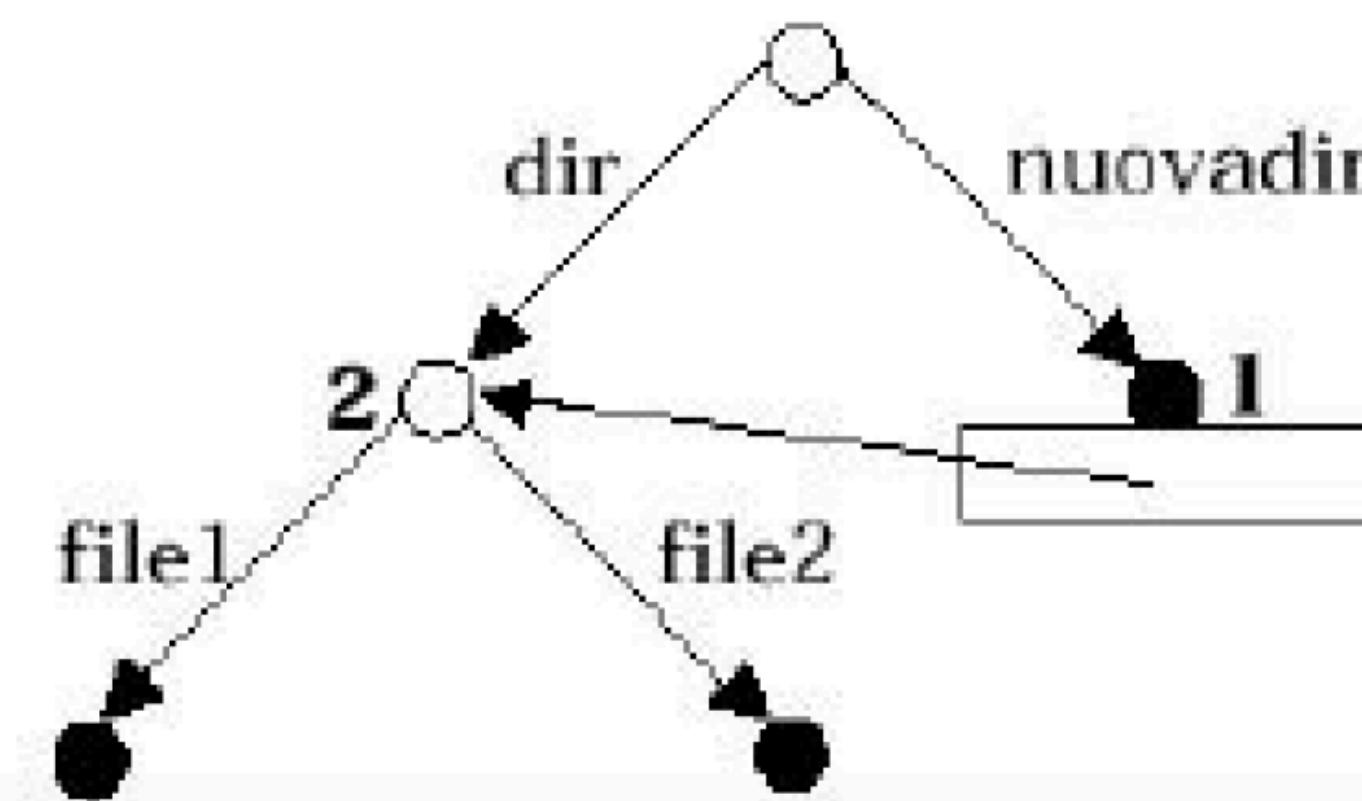
Ln

```
% ln file dir          link a file da directory
% ls -l dir
total 0
-rw-r--r--  3 roberto  usrmail      0 Mar 11 19:40 file
% ln dir nuovissimo
ln: dir is a directory
%
```



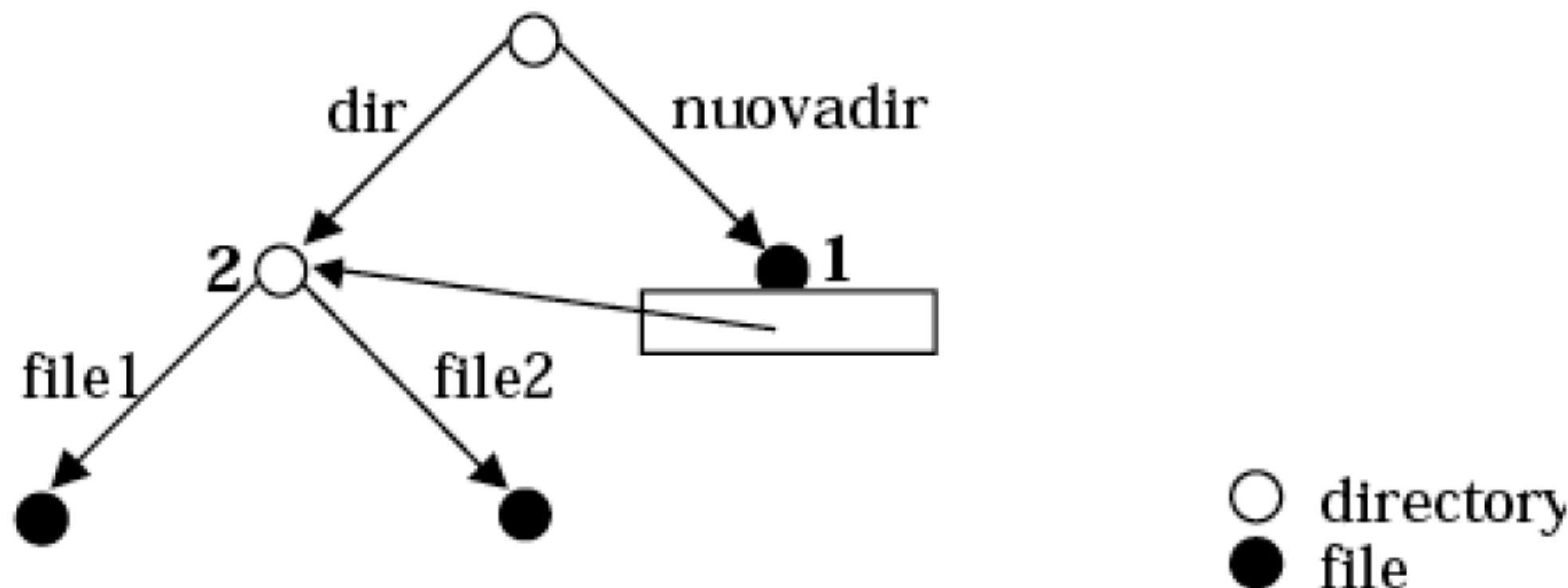
Link simbolici

- ln -s name1 name2
- Permette di creare link a directory;
- Permette di creare link fra file o directory che stanno su file system diversi;
- Viene creato un file name2 che contiene il link simbolico (i.e. il path di name1)



Esempio

```
% ls                               una directory ...
dir
% ls dir                           ... contenuto della directory...
file1  file2
% ln -s dir nuovadir             ...link simbolico a dir da nuovadir
% ls
dir      nuovadir
% ls nuovadir
file1  file2
% ls -l                         ...dir con 2 rif. nuovadir con 1 rif.
total 4
drwxr-sr-x  2 roberto  usrmail  512 Mar 11 19:24
dir
lrwxrwxrwx  1 roberto  usrmail    3 Mar 11 19:24
nuovadir -> dir
%
```



MV

- **mv [options] name...target**
 1. muove il file o directory name sotto la directory target;
 2. se name e target non sono directories, il contenuto di target viene sostituito dal contenuto di name

Esempio

- Caso1:

```
% ls  
file1      file2      targetdir  
% mv file1 file2 targetdir  
% ls  
targetdir  
% ls targetdir  
file1      file2  
% mv targetdir/file1 targetdir/file2 .  
% ls  
file1      file2      targetdir
```

- Caso2:

Se target è un file:

```
% ls  
file1      file2      file3      targetfile  
% mv file1 targetfile  
% ls  
file2      file3      targetfile  
% mv file2 file3 targetfile  
mv: Target targetfile must be a directory  
Usage: mv [-f] [-i] f1 f2  
        mv [-f] [-i] f1 ... fn d1  
        mv [-f] [-i] d1 d2
```

Se target è una directory

Esempio

- Caso3: **Se target non esiste:**

```
% ls
file1      file2
% mv file1 file2 target
mv: target not found
% mv file1 target
% cat target
contenuto di file1
%
```

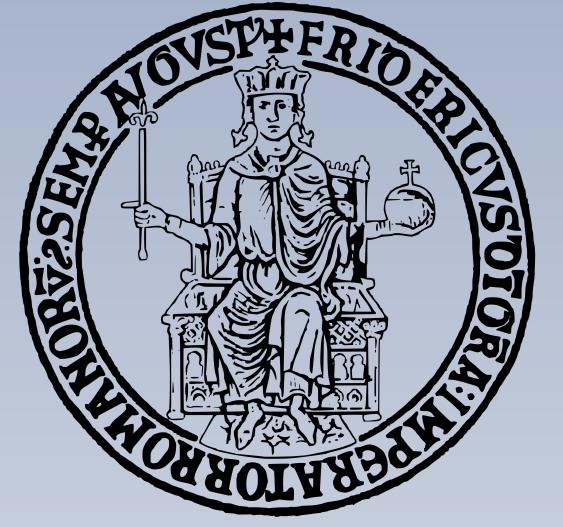
CP

- **cp [options][name...]** target
- come **mv**, ma name viene copiato

```
% ls  
file1 file2 targetdir  
% cp file1 file2 targetdir  
% ls . targetdir  
.:  
file1 file2 targetdir  
  
targetdir:  
file1 file2  
  
% ls  
file1 targetfile  
% cp file1 targetfile  
% ls  
file1 targetfile  
~
```

RM

- **rm [-r] name...**
- rimuove i files indicati
- se un file indicato è una directory: messaggio di errore, a meno che non sia specificata l'opzione -r
 - ... nel qual caso, rimuove ricorsivamente il contenuto della direttrice



Universitá degli Studi di Napoli Federico II

THANK YOU FOR YOUR ATTENTION

TRUST ME ...
I AM A ROBOT!

