



Esercitazione: Introduzione a Linux



Corso di Laurea in Ingegneria Informatica

Università degli Studi di Napoli Federico II

Anno Accademico 2023/2024, Canale San Giovanni



- *Sommario*

- *Cenni storici sui S.O. UNIX e Linux*
- *Linux: shell, file system, comandi di base*

- *Riferimenti*

- *P. Ancilotti, M. Boari, A. Ciampolini, G. Lipari, "Sistemi Operativi", Mc-Graw-Hill (Cap.7)*
- *Dispensa didattica su Unix*



UNIX

...è una famiglia di **SO multiutente e multitasking (time-sharing)**

...nato con l'obiettivo di sostituire i grossi e pesanti sistemi per la gestione di mainframe presenti alla fine degli anni 60.

PRO

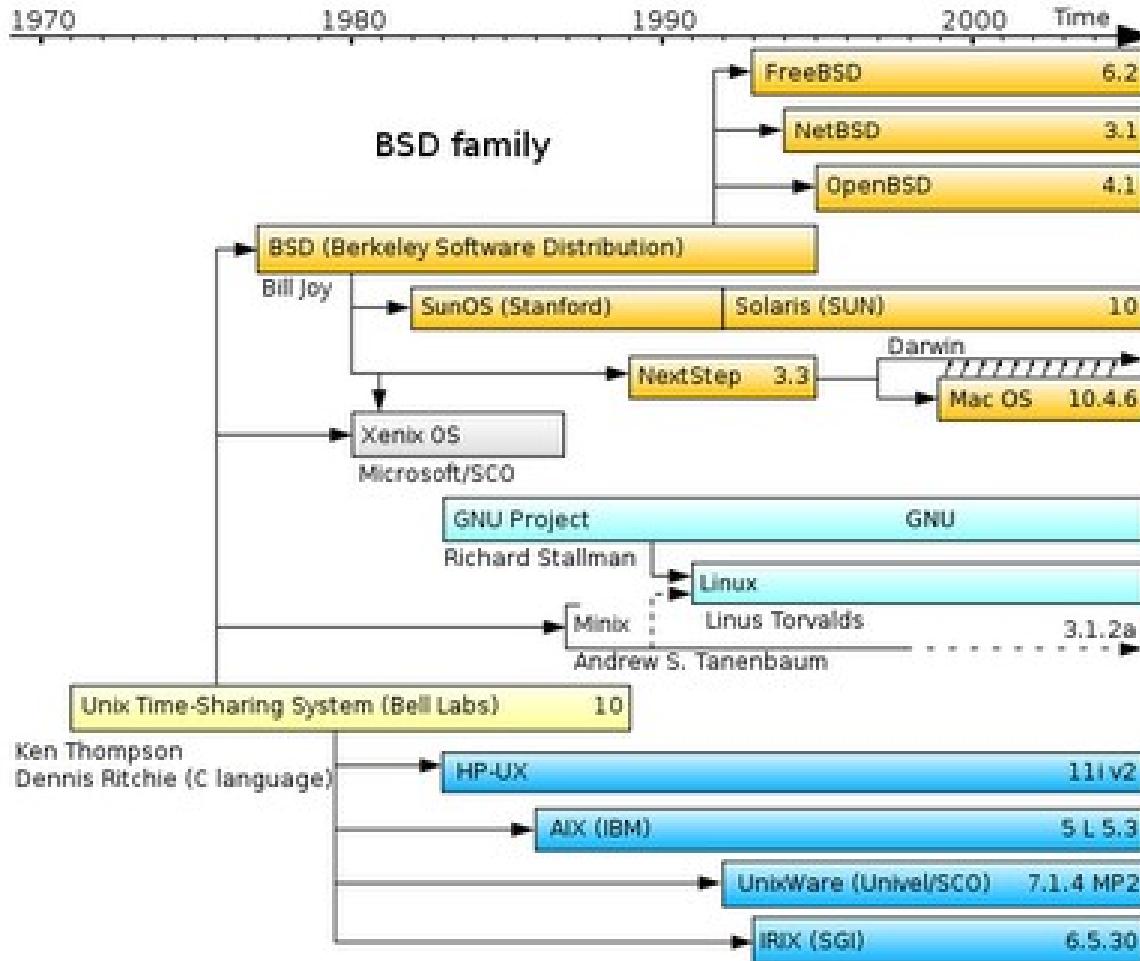
Robustezza, Flessibilità, Portabilità, Multiutenza

CONTRO

Configurazione complessa, interfaccia “shell-oriented”



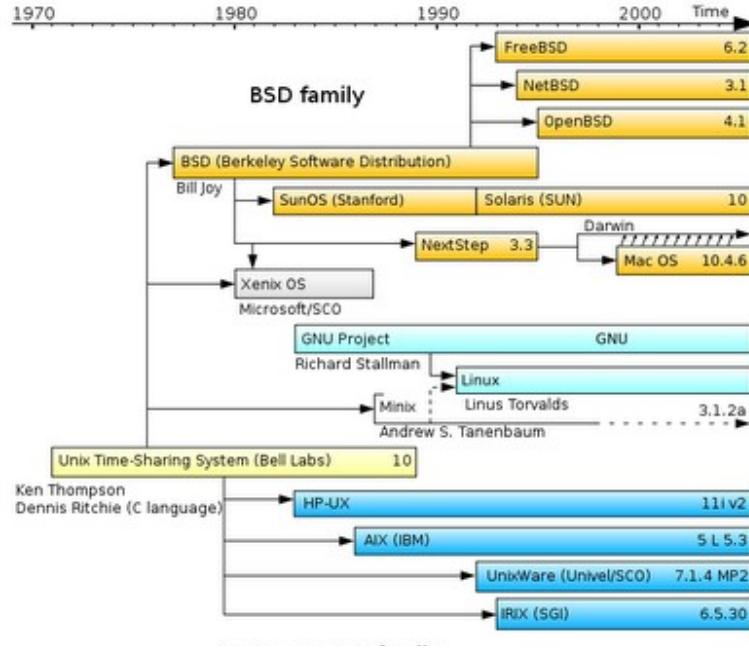
Genealogia UNIX





Genealogia UNIX

< 1970



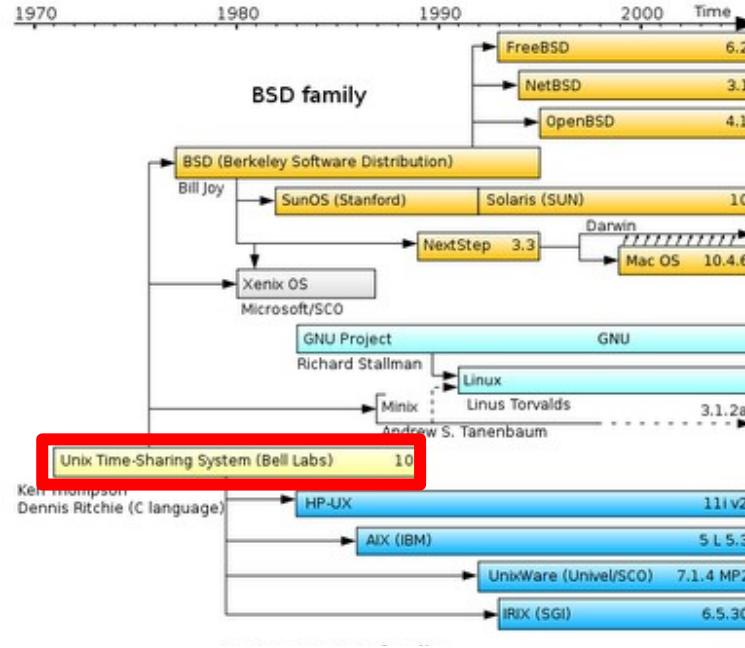
Il **MIT**, in collaborazione con **Bell Labs di AT&T**, sviluppa il SO **MULTICS** (*MULTIplexed Information and Computing Service*). Questo SO è il precursore degli attuali sistemi **time-sharing**.

Il progetto MULTICS **fallisce** a causa della eccessiva complessità.



Genealogia UNIX

1970-1980



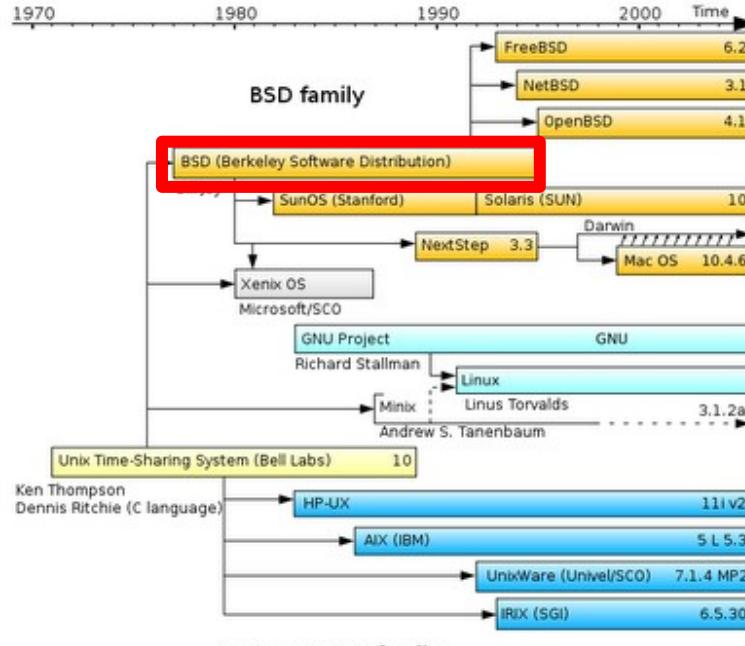
Ken Thompson presso **Bell Labs** realizza un sistema più semplice (per minicomputer **PDP-7**) denominato UNICS, poi rinominato **UNIX**.

In collaborazione con Dennis Ritchie, il sistema UNIX fu riscritto in **linguaggio C**, e portato su altre piattaforme maggiormente popolari, tra cui DEC e PDP-11 (piccoli mainframe diffusi nelle università).



Genealogia UNIX

1980-1985



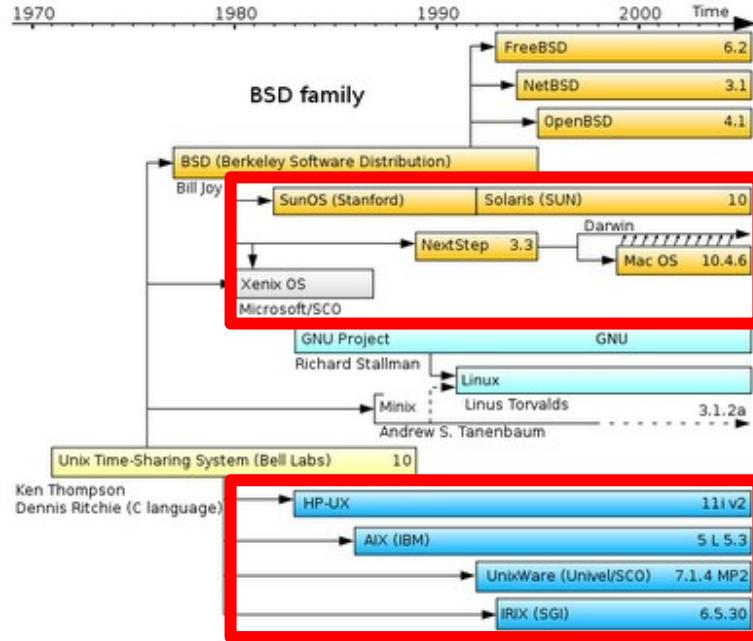
AT&T (che inizialmente non poteva commercializzare UNIX) concede il codice sorgente di UNIX in licenza a varie università, che sviluppano nuove estensioni.

In particolare, **BSD** (*Berkeley Software Distribution*) introdusse la **memoria virtuale**, la **paginazione**, varie migliorie al **filesystem**, e il supporto ai protocolli implementati nelle reti **TCP/IP**.



Genealogia UNIX

1985-1990



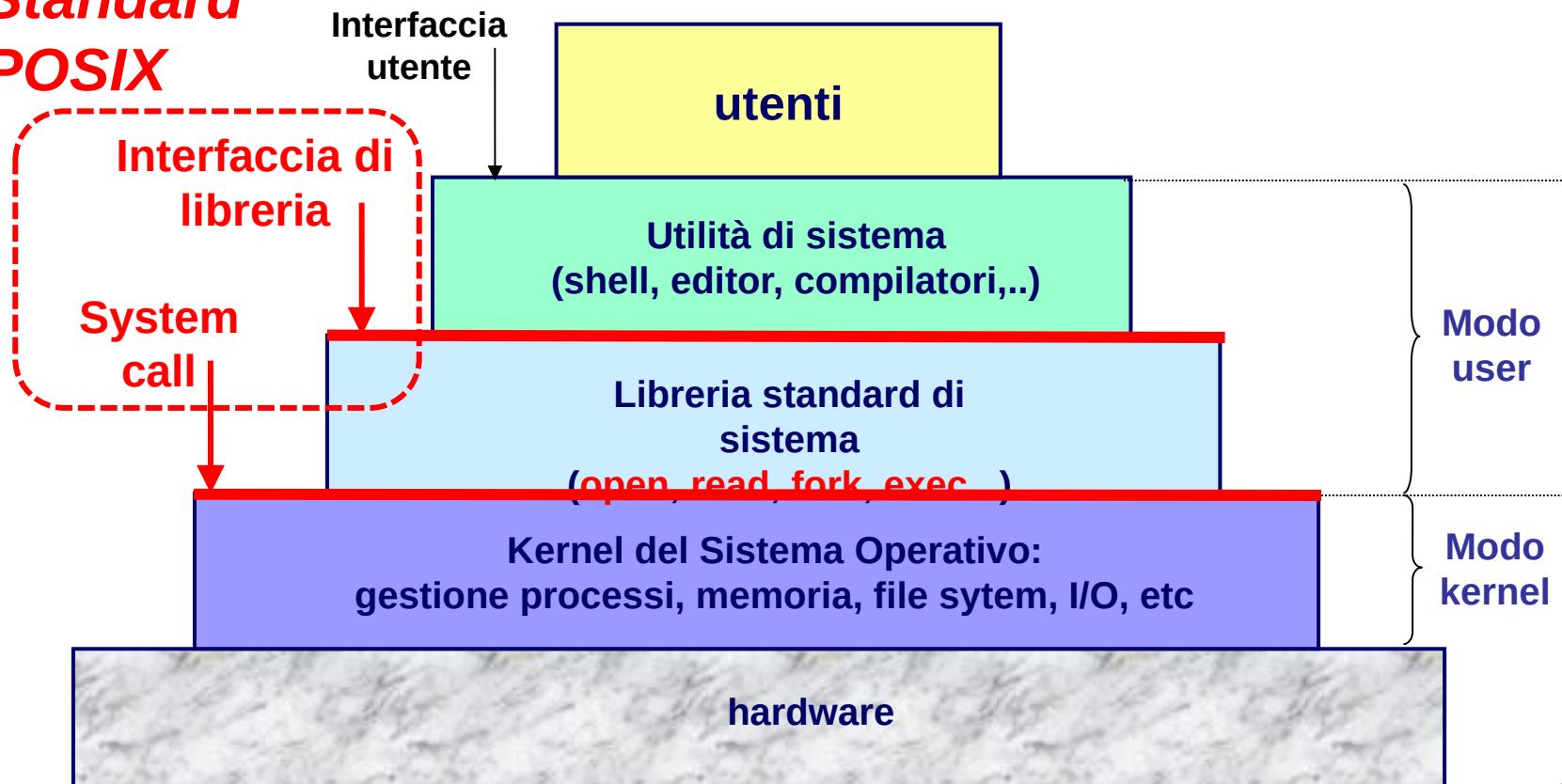
AT&T (scissa nel 1984) e altri vendor commercializzano una versione di UNIX, **System V**, sviluppata parallelamente a BSD.

Per favorire la portabilità delle applicazioni tra le varianti di UNIX, nel 1988 l'**IEEE** definisce lo standard **POSIX**, che stabilisce (dall'intersezione di System V e BSD) un'interfaccia di **libreria standard** per le applicazioni.



Unix: architettura a livelli

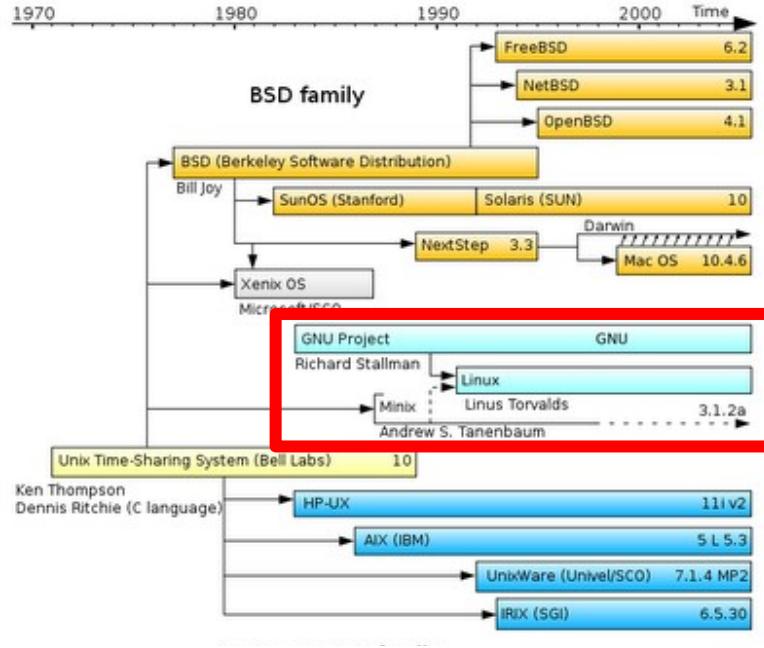
**Standard
POSIX**





Genealogia UNIX

>1990



Linus Torvalds (studente dell'Università di Helsinki, Finlandia) sviluppa **Linux**, un SO **POSIX-compatibile** per processori **Intel 386**, che diventa lo “UNIX open-source” di riferimento di un'ampia comunità di sviluppatori.



Linux: la sua storia



Linus Benedict Torvalds



Hello everybody out there using minix -

I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).

I've currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :-)

Linus (torv...@kruuna.helsinki.fi)

PS. Yes - it's free of any minix code, and it has a multi-threaded fs. It is NOT portable (uses 386 task switching etc), and it probably never will support anything other than AT-harddisks, as that's all I have :-(.





Linux: le distribuzioni

- Il *kernel Linux* è un singolo progetto
- Negli anni, sono proliferate molte *distribuzioni* di questo sistema operativo (free e commerciali)
- Una distribuzione include
 - *kernel*
 - *applicazioni* (browser, editor, compilatori, ...)
 - *strumenti* per l'amministrazione di sistema



MkLinux

Linux: le distribuzioni

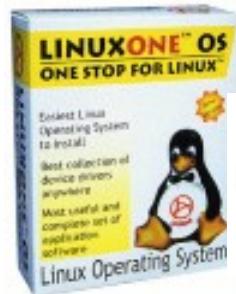


LinuxPPC



Linux Pro™

fedora^f



Linux Mandrake™

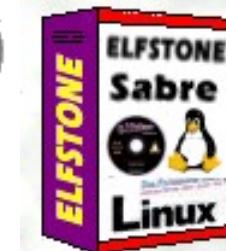
Platinum
LINUX



redhat

debian

Redmondlinux
operating system



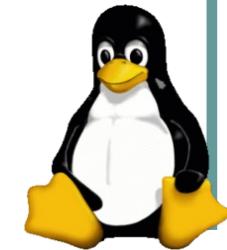
DILX



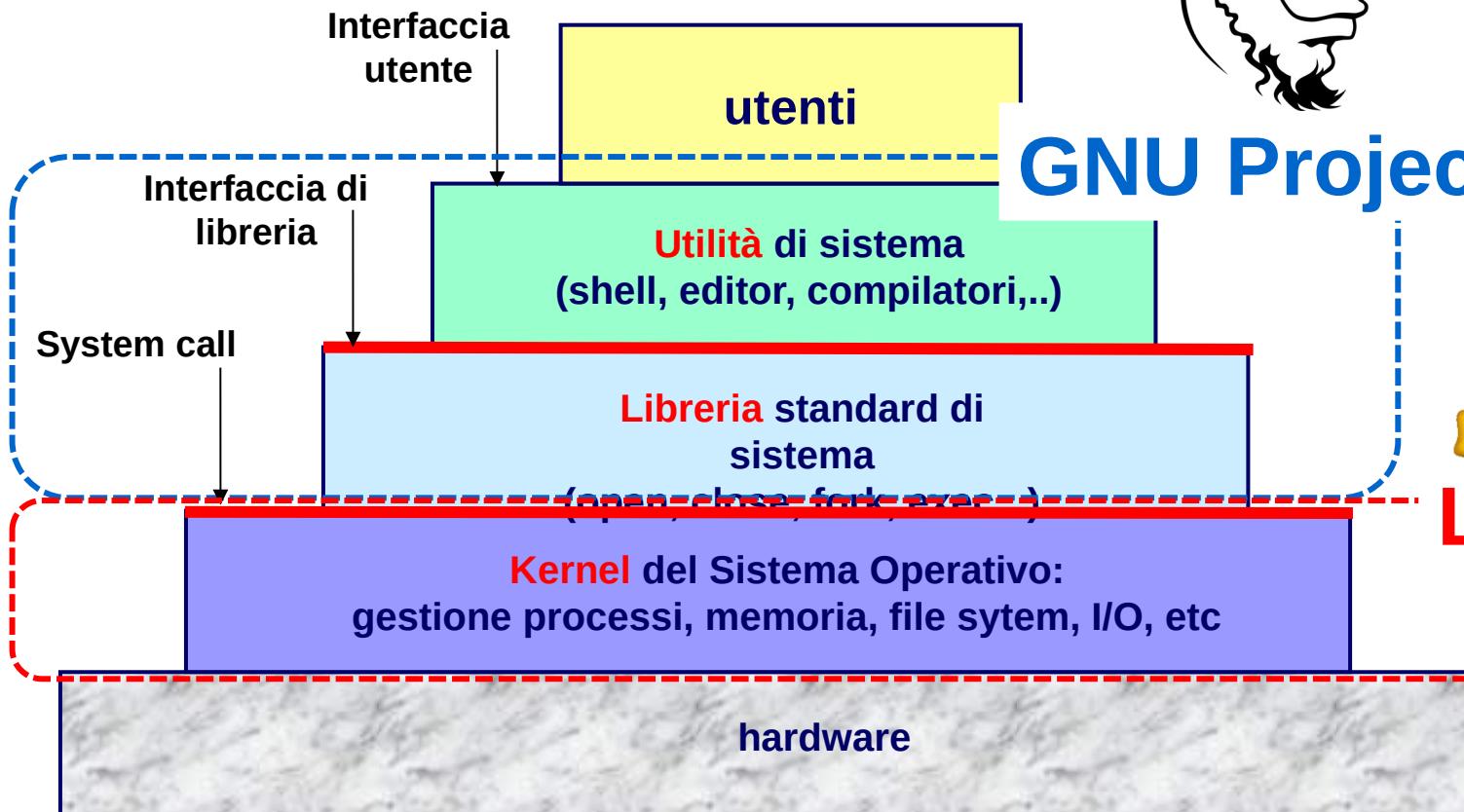
GNU/Linux



GNU Project



Linux





Il concetto di “free software”

- Linux è un *free software* (software libero)
- Distribuito con licenza *GNU General Public License (GPL)*
- Da non confondere con semplicemente "gratis" o "di pubblico dominio"!





Il concetto di “free software”

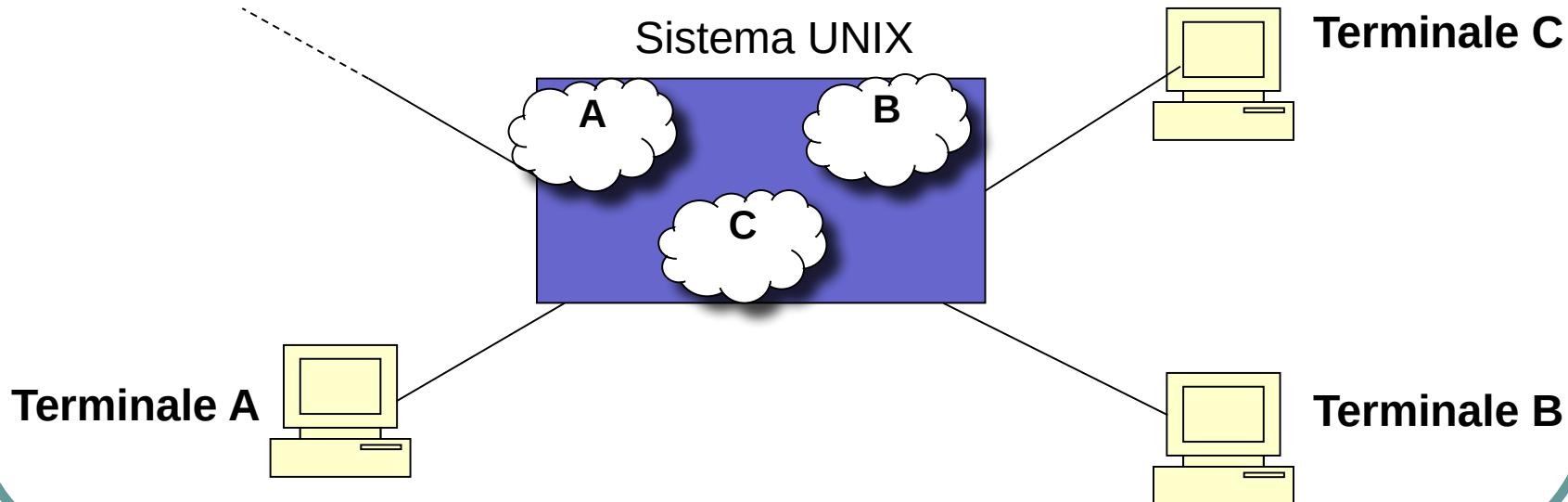
Il software libero pone dei vincoli di uso, come:

1. *se si modifica il programma, gli **autori precedenti della versione originale devono essere menzionati**, lasciando intatto il loro copyright*
2. *non è consentito re-distribuire il programma **senza anche distribuire il codice sorgente**, con le eventuali modifiche*
3. *non è consentito re-distribuire il programma con una **licenza d'uso non-libera***



Linux: caratteristiche generali

- Come *UNIX*, *Linux* è un sistema:
 - Multiutente
 - Multiprogrammato (*time sharing*)
 - Interattivo





Il terminale





Il terminale

- *Il "terminale" è un dispositivo combinato display/tastiera*
- *Non è un computer a sé stante! Era usato come "client" nei primi sistemi*
- *Interfaccia "a caratteri"*
- *Anche detto "console", "TeleTYpewriter"*





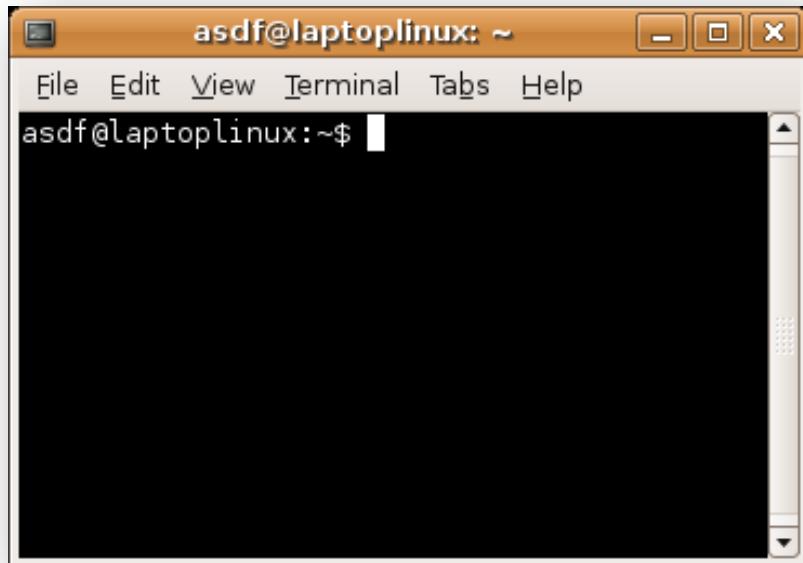
Il terminale





La shell

- *La **shell** (“guscio”, “invólucro”) è un programma che interagisce con l’utente tramite **comandi testuali***



Emulatore di terminale

(simula la console tradizionale in una finestra grafica)

- *Avviare e gestire processi*
- *Operare sul filesystem
(creazione, copia, spostamento file)*
- *Amministrazione di sistema
(aggiornamenti, backup, networking, etc.)*



La shell

```
tizio@computer:~$ comando ↵ INVIO  
... viene eseguito il comando ...  
... risultati stampati a video ...  
  
tizio@computer:~$
```

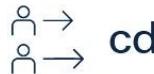
Il carattere **"\$"** indica che la shell è in attesa che l'utente digitи un comando (prompt)



Comandi UNIX shell



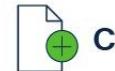
pwd
To find out the path
of the current
working directory



cd
To navigate through
the Linux files and
directories



ls
Is used to view
the contents of
a directory



cat
To view the
contents of a
file



cp
To copy files from
the current directory
to a different directory



mv
The command
is to move files



mkdir
Use mkdir command
to make a new
directory



rmdir

The rm command
is used to delete
directories and
the contents
within them



locate
You can use this
command to locate
a file, just like the search
command in Windows



sudo
This command enables
you to perform tasks
that require administrative
or root permissions



head
The head command
is used to view the
first lines of any text



Comandi UNIX shell

ls

Elenca i file che sono dentro una cartella

cp

Copia un file

mv

Sposta un file

mkdir

Crea una nuova cartella

touch

Crea un nuovo file (vuoto)

rm

Rimuove un file

cat

Stampa a video il contenuto di un file

...

...



Comandi UNIX shell

I comandi necessitano di **parametri di ingresso**, sotto forma di **stringhe** di caratteri separate da spazi

cp /persorso/file /percorso/destinazione

↑
comando

↑
parametri,
in questo caso:
1. *Un file esistente sul disco*
2. *Un file-copia da creare*



UNIX shell

Alcuni comandi accettano come parametri (opzionali) delle **stringhe prefissate** (dette “**flag**”) che iniziano per “-”

ls

stampa a video la **lista dei file** nella cartella di lavoro corrente

ls -l

stampa a video la lista dei file, **aggiungendo informazioni** sui permessi, dimensione, tipo, etc.

ls -a

stampa a video la lista dei file, **includendo i cosiddetti “file nascosti” (iniziano per “.”)** che non sono normalmente elencati

ls -a -l

tre **alternative equivalenti** per stampare sia le informazioni aggiuntive, sia per includere i file nascosti

ls -l -a

ls -la



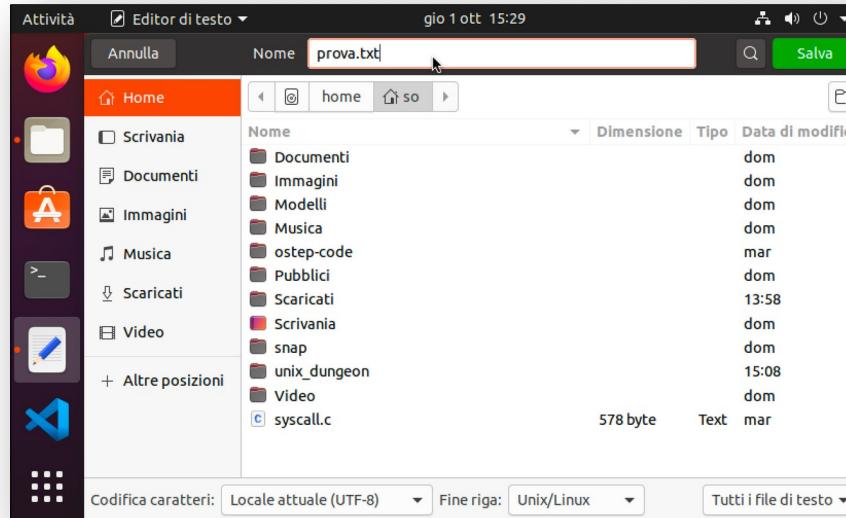
Percorsi assoluti in UNIX

- In modo simile a Windows, i file in Linux sono identificati da un **percorso assoluto**
- Un file nella cartella dell'utente in Windows è:
C:\Users\roberto\ciao.txt
- In Linux diventa:
/home/roberto/ciao.txt
- Sequenza di nomi cartelle (a partire dalla "radice"), separate da **"/"**, più il nome del file



Percorsi assoluti in UNIX

- Ad esempio, se con l'editor di testo creiamo un file "*prova.txt*" e lo salviamo nella cartella dell'utente, il suo percorso sarà:
/home/utente/prova.txt



- Possiamo stampare il file dal terminale, digitando:

```
cat /home/utente/prova.txt
```



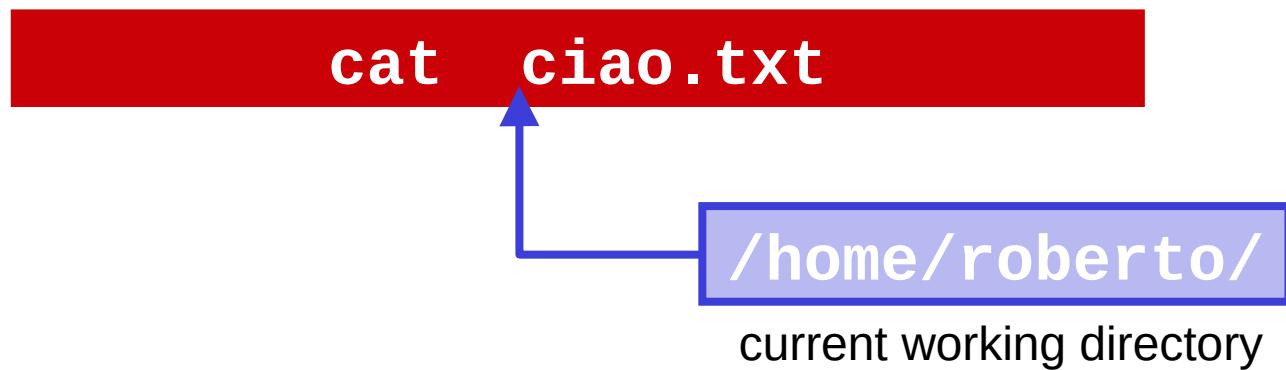
Percorsi relativi

- *Nella shell è scomodo dover digitare per intero il percorso assoluto dei file*
- *I percorsi relativi sono una versione "breve"*
- *E.s. il comando (percorso assoluto):*
cat /home/roberto/ciao.txt
- *è uguale a questo comando (p. relativo):*
cat ciao.txt



Percorsi relativi

- La shell riconosce un percorso relativo *dalla assenza dello "/" iniziale*
- La shell aggancia al percorso quello della "*current working directory*" (*CWD*)





Muoversi tra le cartelle

~ \$

All'avvio della shell, la “*current working directory*” è inizialmente `/home/user/` (abbreviato con `~`)
\$ è il simbolo del cursore dei comandi

~ \$ `cat prova.txt`

`/home/user/prova.txt`

~ \$ `cd Downloads`

Il comando `cd` modifica la *current working directory*, che diventa `/home/user/Downloads`

~/Downloads \$

`cat prova.txt`

`/home/user/Downloads/prova.txt`



Percorsi assoluti e relativi

```
~/Downloads $ mv ciao.txt ..//ciao.txt
```

/home/user/Downloads/prova.txt

Il simbolo " .. " (doppio punto) rappresenta la cartella "padre" della CWD.

Il percorso diventa:
/home/user/prova.txt

Nota: il simbolo " . " (punto) rappresenta la CWD.

Il " ." è necessario in alcuni casi, lo useremo più avanti



Un'introduzione didattica alla shell

Screenshot of a GitHub repository page for "rnatella / unix_dungeon".

The repository has 1 branch and 0 tags. The master branch contains files: "ingresso", "LICENSE", and "README.md".

The "Code" dropdown menu is open, showing options: Go to file, Add file, and Clone. The "Clone" section displays the URL: https://github.com/rnatella/unix_dungeon.

The "About" section states: "No description, website, or topics provided." It includes links for "Readme" and "CCO-1.0 License".

The "Releases" section indicates "No releases published" and provides a link to "Create a new release".

The "Packages" section indicates "No packages published".

Introduzione

In questo gioco, impersonerai un aspirante mago che affronta la sua prova di iniziazione.

https://github.com/rnatella/unix_dungeon/



Un'introduzione didattica alla shell

```
~ $ git clone https://github.com/rnatella/unix_dungeon  
~ $ cd unix_dungeon  
~/unix_dungeon $ cd ingresso  
~/unix_dungeon/ingresso $ cat leggimi  
Ti trovi in un sotterraneo buio.  
...  
...
```



Comandi UNIX

- Quando digitiamo "cat", "ls", etc., stiamo indicando alla shell un programma
- La shell trova automaticamente il *percorso del programma* nelle **cartelle di sistema**
 - /bin, /sbin, /usr/bin, /usr/local/bin, ...



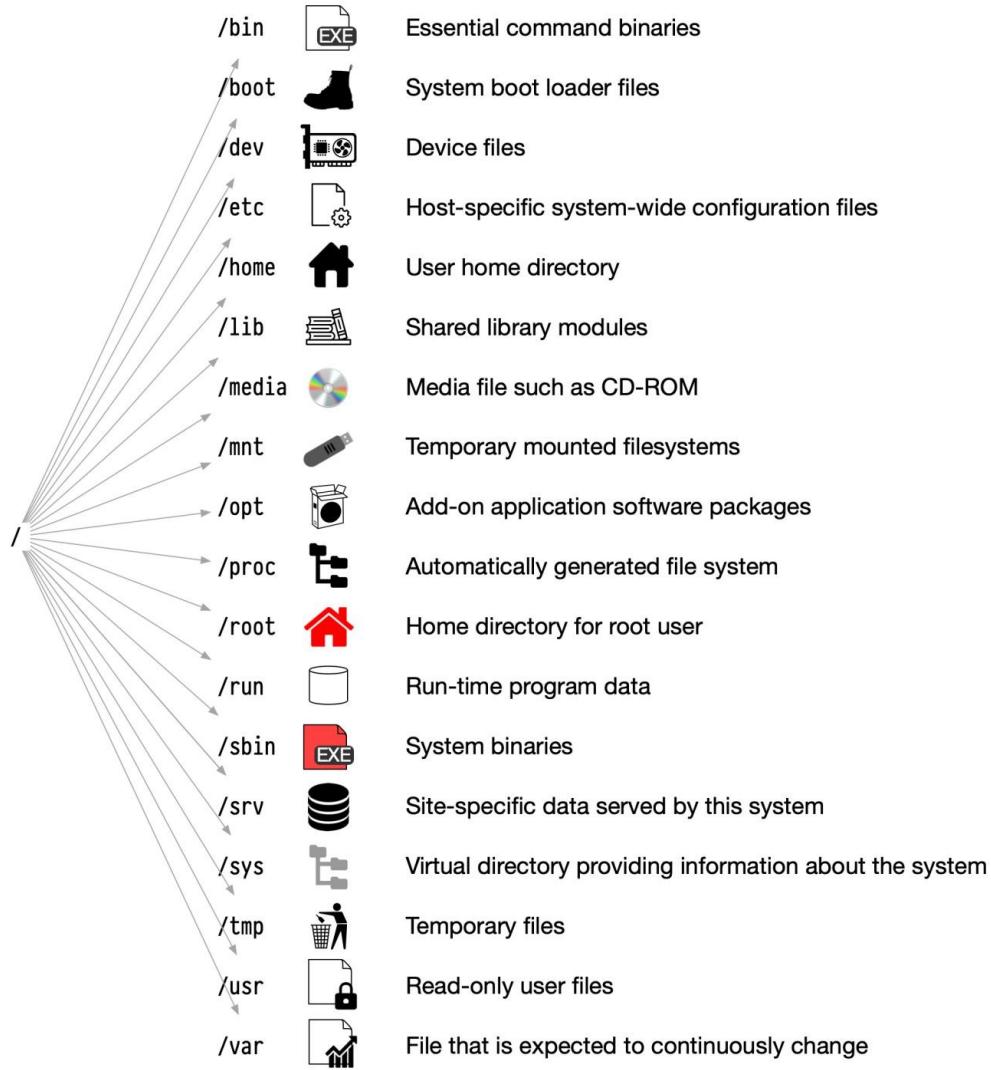
Nota: In Linux, i programmi eseguibili non hanno l'estensione «.exe» come in Windows



Filesystem tipico in Linux

Linux File Systems

ByteByteGo.com





Esecuzione di programmi

- Per lanciare un *nostro programma* (invece di quelli di sistema), si deve scrivere il suo percorso

```
~ $ gcc programma.c -o mioexec  
~ $ /home/utente/mioexec
```

- Per brevità, si usano percorsi relativi, es.

```
~ $ ./mioexec
```

Il `./` iniziale è importante! Indica alla shell che il programma si trova nella CWD (non in /bin o altrove)



Esecuzione di programmi

```
so@so-vbox:~/Prova$ ls -l
totale 24
-rwxrwxr-x 1 so so 16688 ott 1 16:17 prova
-rw-rw-r-- 1 so so     72 ott 1 16:16 prova.c
```

```
so@so-vbox:~/Prova$ prova
Comando «prova» non trovato, si intendeva forse:
comando «prove» da deb perl (5.30.0-9build1)
comando «prove» da deb libtest-harness-perl (3.42-2)
comando «proda» da deb proda (1.0-12build1)

Provare: sudo apt install <nome deb>
```

```
so@so-vbox:~/Prova$ ./prova
Hello World
so@so-vbox:~/Prova$
```

ERRATO:

*La shell cerca un programma
di nome "prova" **nelle cartelle
di sistema**
(es. /bin/prova)*

CORRETTO:

*Con "./", la shell tratta il token
come **percorso relativo**,
trova: /home/so/Prova/prova*



Esecuzione di programmi

```
so@so-vbox:~$ ls
Documenti  Musica      Pubblici   snap          Video
Immagini   ostep-code  Scaricati   syscall.c
Modelli     Prova       Scrivania  unix_dungeon
so@so-vbox:~$
```

Anche "ls" è un **programma eseguibile**.

Senza "/", la shell cerca il programma in **/bin/ls**

```
so@so-vbox:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/
usr/games:/usr/local/games:/snap/bin
so@so-vbox:~$
```

Elenco delle cartelle con gli eseguibili di sistema



Utenti e gruppi

```
$ ls -l
-rwxrwxr-x 1 so so 16696 gen 28 19:44 prova
-rw-rw-r-- 1 so so      79 gen 28 19:44 prova.c
```

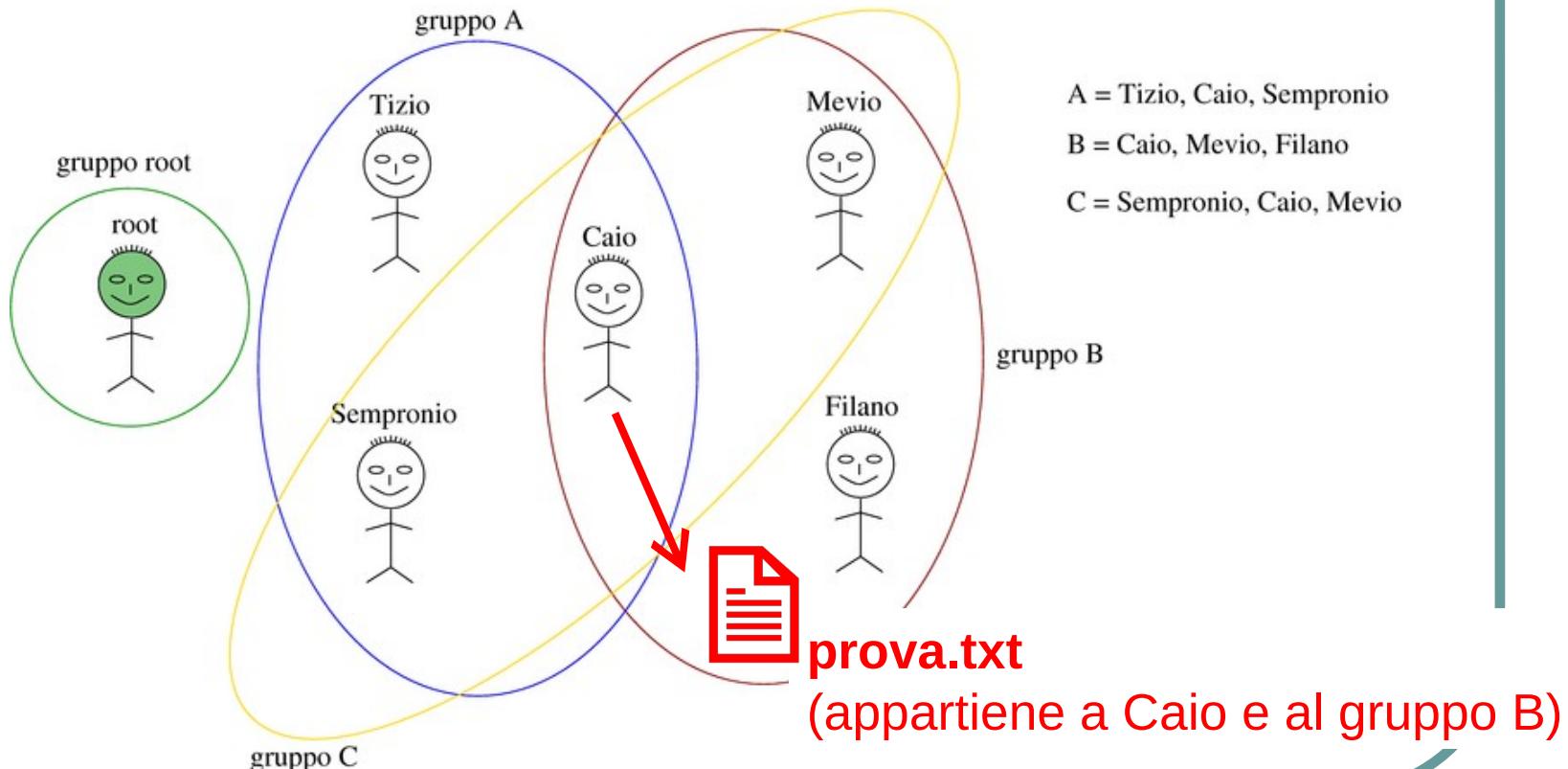
L'utente "so" è il creatore e proprietario dei file

- In Linux, ogni file ha un "*utente proprietario*"
- Di default, è l'utente che ha creato il file
- Si può impedire agli altri utenti del computer di leggere o cambiare il file



Utenti e gruppi

- Linux permette di dividere gli utenti in *gruppi*
- Utile per la *condivisione* (selettiva) di file





Utenti e gruppi

```
$ ls -l
-rwxrwxr-x 1 so so 16696 gen 28 19:44 prova
-rw-rw-r-- 1 so so      79 gen 28 19:44 prova.c
```

Il gruppo (fittizio) "so" ha accesso al file

- Alla creazione di un utente, viene creato un *gruppo omonimo* a lui riservato
- I file di "*tizio*" appartengono anche al gruppo "*tizio*"



Il comando chown

```
chown utente:gruppo nomefile
```

- Modifica l'utente e il gruppo proprietari del file
- Ad esempio:

```
chown so:sviluppo miofile.txt
```

tutti gli utenti del gruppo "sviluppo" sono proprietari del file



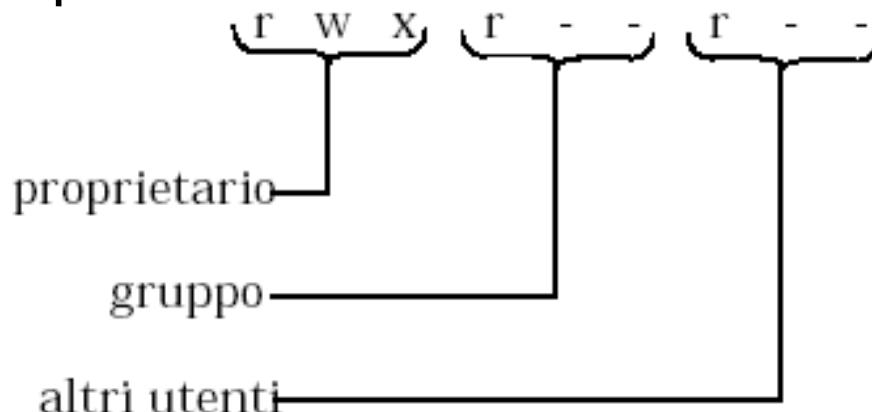
Permessi

- Ad un file possono essere attribuiti i seguenti permessi:

r : readable
w : writable
x : executable

} per {
proprietario
gruppo
altri utenti

Esempio:

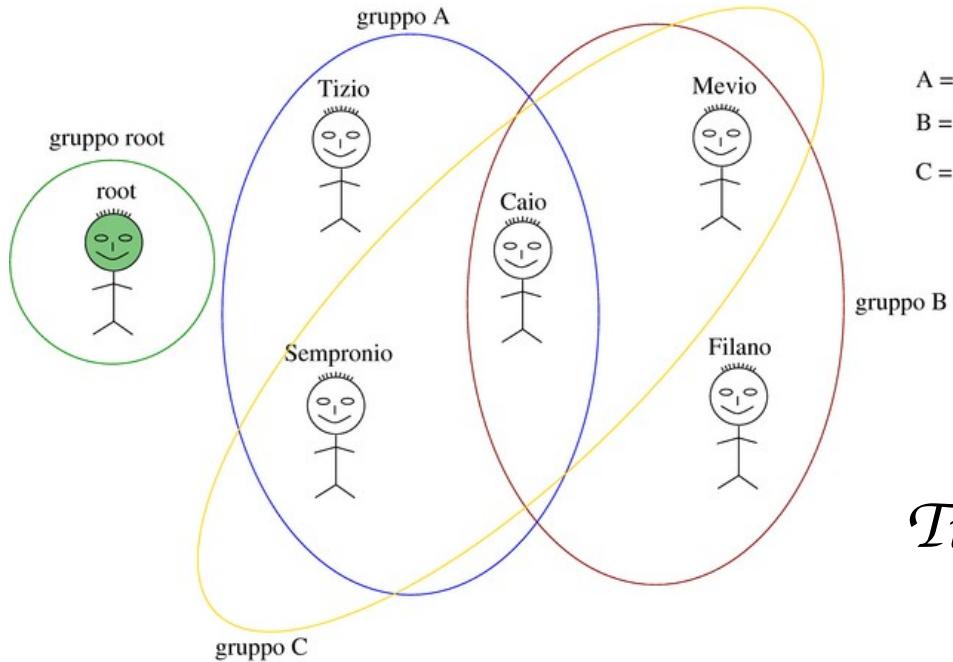


In binario: 111 100 100

In ottale: 7 4 4



Utenti e gruppi



A = Tizio, Caio, Sempronio

B = Caio, Mevio, Filano

C = Sempronio, Caio, Mevio

*Quali file può leggere
Tizio? Quali può scrivere?*

```
-rw-r----- 1 Tizio GruppoA ... file1
-rw-r----- 1 Caio GruppoA ... file2
-rw-r----- 1 Caio GruppoB ... file3
```



Il comando chmod

chmod permissions nomefile

- Assegna i permission al file nomefile
- Può essere eseguito solo dal proprietario del file (o dall'amministratore)
- Ad esempio (versione in ottale):

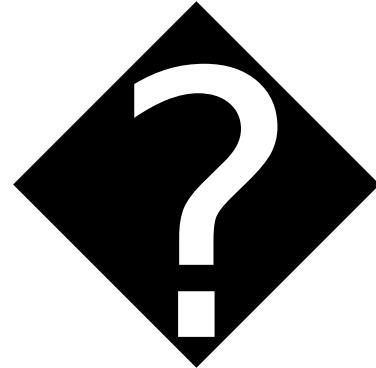
chmod 644 miofile.txt

assegna i permessi rw- r-- r--



Utente "root"

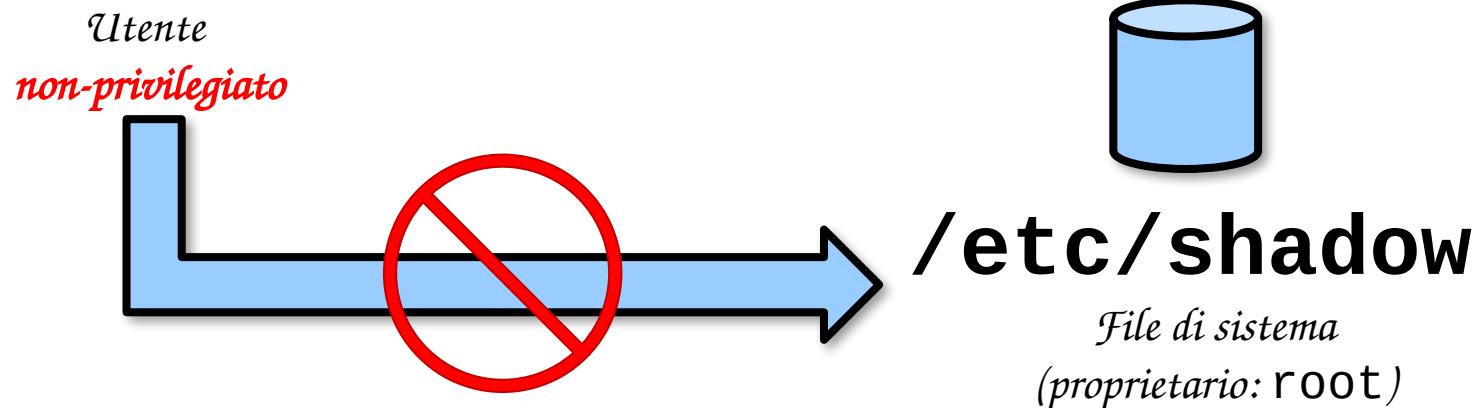
- In *UNIX*, l'utente **root** (superuser) ha privilegi speciali
 - *accesso a tutto il filesystem*
 - *Accesso a tutte le periferiche*
 - *creazione e cancellazione degli utenti*
 - ...





Comandi privilegiati

- Anche gli utenti non-privilegiati hanno bisogno di accedere a file/cartelle riservati
- Esempio: aggiornare la password





Comandi privilegiati

- Due possibilità: *su* e *sudo*
- Comando *su* :
Avvia una nuova shell, come utente privilegiato ("root")
- Comando *sudo* :
Esegue un singolo comando come utente privilegiato, la shell rimane non-privilegiata



Utenti privilegiati

- Il comando ***su*** (“switch user”) modifica i privilegi della shell in cui viene avviato

****su [nome utente]****

A screenshot of a terminal window titled "root@so-vb...". The window shows the command "su" being run by the user "so" at the prompt "so@so-vbox:~\$". A password prompt "Password:" appears, followed by the output of the "whoami" command showing the user is now "root".

```
root@so-vb... ~$ su
Password:
root@so-vbox:/home/so# whoami
root
root@so-vbox:/home/so#
```



Utenti privilegiati

- Il comando **sudo** permette di eseguire **singoli comandi** come se si fosse un altro utente

sudo [comando]

```
so@so-vbox:~$ sudo tail -5 /etc/shadow
pulse:*:18474:0:99999:7:::
gnome-initial-setup:*:18474:0:99999:7:::
gdm:*:18474:0:99999:7:::
so:$6$TUj5dT25Ubaw/KAL$66YXVt2bSqSFClI5JdFLmg6.6jei5FSdzbZjy1WXj
2dhnskXtBJNddOBGNswzM2iqSZw3pgCuBXAd.jtH88h1/:18532:0:99999:7:::
systemd-coredump:!!!:18532:::::
so@so-vbox:~$
```



setuid

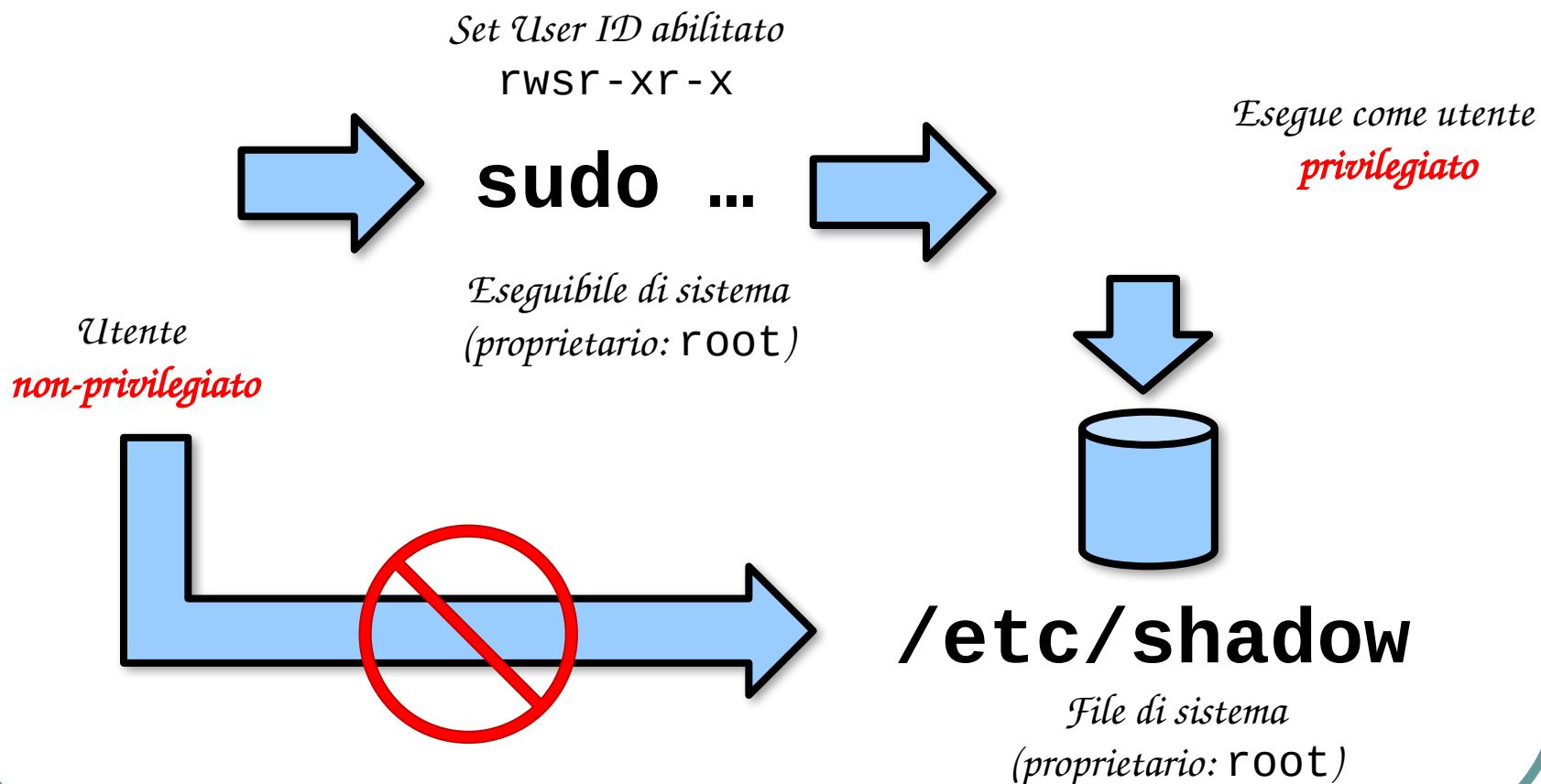
- *su e sudo sono eseguibili setuid (set user id)*
- *Il programma esegue con i privilegi dell'utente proprietario del file eseguibile*

```
so@so-vbox:~$ ls -l /bin/sudo
-rwsr-xr-x 1 root root 166056 gen 19 15:21 /bin/sudo
so@so-vbox:~$ ls -l /bin/su
-rwsr-xr-x 1 root root 67816 lug 21 2020 /bin/su
so@so-vbox:~$
```

Quando si lancia su/sudo, il processo viene *elevato ad amministratore*



setuid





setuid

Vulnerabilità nel comando SUDO in Linux consente
escalation di privilegi

05/02/2020

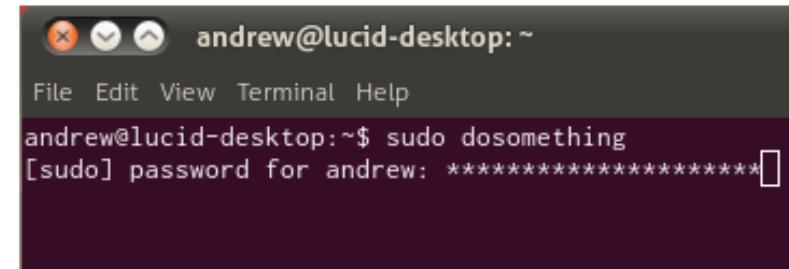
cve-2019-18634 | Linux | Vulnerabilità



Un ricercatore Apple ha scoperto una vulnerabilità nella utility *sudo*, utilizzata dai sistemi Linux e MacOs, per lanciare comandi con privilegi elevati (root) nei sistemi impattati.



Content
[Weekly Edition](#)
[Archives](#)
[Search](#)
[Kernel](#)
[Security](#)
[Distributions](#)
[Events calendar](#)
[Unread comments](#)



```
andrew@lucid-desktop:~$ sudo dosomething
[sudo] password for andrew: *****
```

User: Password: |

A major vulnerability in Sudo

A major vulnerability in Sudo

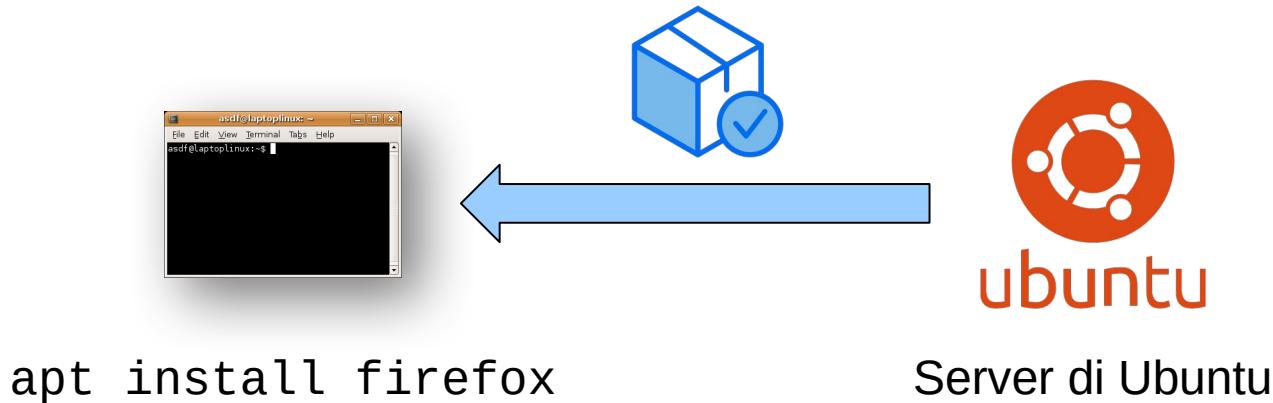
[Security] Posted Feb 3, 2021 1:03 UTC (Wed) by jake

A longstanding hole in the [Sudo](#) privilege-delegation tool that was [discovered in late January](#) is a potent local vulnerability. Exploiting it allows local users to run code of their choosing as root by way of a bog-standard heap-buffer overflow. It seems like the kind of bug that might have been found earlier via code inspection or fuzzing, but it has remained in this security-sensitive utility since it was introduced in 2011.



Installazione del software

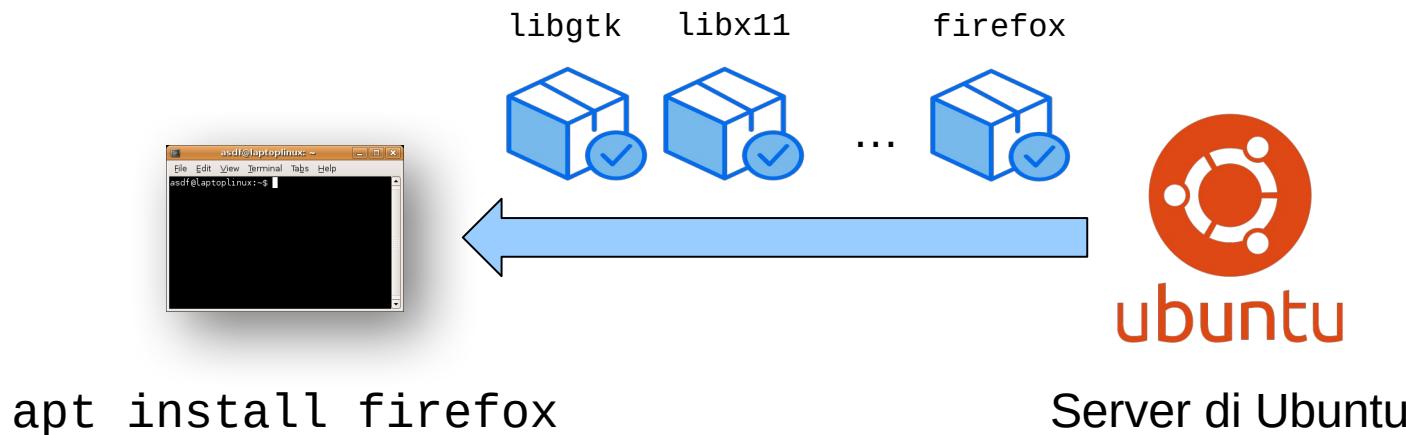
- Le distribuzioni Linux danno la possibilità di installare **pacchetti pre-compilati** di software **open-source**
 - Debian, Ubuntu e derivate hanno il comando **apt**
 - Red Hat, Fedora, e CentOS usano **yum**





Installazione del software

- Questi tool scaricano automaticamente anche eventuale altro software aggiuntivo di cui il programma necessita (**dipendenze**)





Comandi principali

- Aggiornare il database dei pacchetti:
 - *sudo apt update*
- Installare un pacchetto
 - *sudo apt install <nomepacchetto>*
- Cercare un pacchetto, il cui nome o descrizione contenga una stringa
 - *apt search <stringa>*



Esempi di uso di apt

```
so@so-vbox:~$ sudo apt update
Trovato:1 http://it.archive.ubuntu.com/ubuntu focal InRelease
Trovato:2 http://it.archive.ubuntu.com/ubuntu focal-updates InRelease
Trovato:3 http://it.archive.ubuntu.com/ubuntu focal-backports InRelease
Scaricamento di:4 http://security.ubuntu.com/ubuntu focal-security InRelease [107 kB]
Scaricamento di:5 http://security.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metadata [24,3 kB]
Scaricamento di:6 http://security.ubuntu.com/ubuntu focal-security/universe amd64 DEP-11 Metadata [55,6 kB]
Recuperati 187 kB in 1s (140 kB/s)
Lettura elenco dei pacchetti... Fatto
Generazione albero delle dipendenze
Lettura informazioni sullo stato... Fatto
6 pacchetti possono essere aggiornati: eseguire "apt list --upgradable" per vederli.
so@so-vbox:~$ 
```



Esempi di uso di apt

```
so@so-vbox:~$ sudo apt install vlc
Lettura elenco dei pacchetti... Fatto
Generazione albero delle dipendenze
Lettura informazioni sullo stato... Fatto
Il seguente pacchetto è stato installato automaticamente e non è più richiesto:
  libfprint-2-tod1
Usare "sudo apt autoremove" per rimuoverlo.
I seguenti pacchetti aggiuntivi saranno inoltre installati:
  libaribb24-0 libbasicusageenvironment1 libcddb2 libdc1394-22 libdca0 libdouble-conversion3 libdvbpsi10 libebml4v5 libfaad2
  libgroupsock8 libxml10 libkate1 liblirc-client0 liblivemedia77 liblua5.2-0 libmad0 libmatroska6v5 libmpcdec6
  libopenmpt-modplug1 libpcre2-16-0 libplacebo7 libprotobuf-lite17 libproxy-tools libqt5core5a libqt5dbus5 libqt5gui5
  libqt5network5 libqt5svg5 libqt5widgets5 libqt5x11extras5 libresid-builder0c2a libsdl-image1.2 libsdl1.2debian libsidplay2
  libsndio7.0 libspatialaudio0 libsrt1 libssh2-1 libupnp13 libusageenvironment3 libvlc-bin libvlcc5 libvlccore9
  libxcb-xinerama0 libxcb-xinput0 qt5-gtk-platformtheme qttranslations5-l10n vlc-bin vlc-data vlc-l10n vlc-plugin-base
  vlc-plugin-notify vlc-plugin-qt vlc-plugin-samba vlc-plugin-skins2 vlc-plugin-video-output vlc-plugin-video-splitter
  vlc-plugin-visualization
Pacchetti suggeriti:
  lirc qt5-image-formats-plugins qtwayland5 sndiod libdvdcss2
I seguenti pacchetti NUOVI saranno installati:
  libaribb24-0 libbasicusageenvironment1 libcddb2 libdc1394-22 libdca0 libdouble-conversion3 libdvbpsi10 libebml4v5 libfaad2
  libgroupsock8 libxml10 libkate1 liblirc-client0 liblivemedia77 liblua5.2-0 libmad0 libmatroska6v5 libmpcdec6
  libopenmpt-modplug1 libpcre2-16-0 libplacebo7 libprotobuf-lite17 libproxy-tools libqt5core5a libqt5dbus5 libqt5gui5
  libqt5network5 libqt5svg5 libqt5widgets5 libqt5x11extras5 libresid-builder0c2a libsdl-image1.2 libsdl1.2debian libsidplay2
  libsndio7.0 libspatialaudio0 libsrt1 libssh2-1 libupnp13 libusageenvironment3 libvlc-bin libvlcc5 libvlccore9
  libxcb-xinerama0 libxcb-xinput0 qt5-gtk-platformtheme qttranslations5-l10n vlc-bin vlc-data vlc-l10n vlc-plugin-base
  vlc-plugin-notify vlc-plugin-qt vlc-plugin-samba vlc-plugin-skins2 vlc-plugin-video-output vlc-plugin-video-splitter
  vlc-plugin-visualization
0 aggiornati, 59 installati, 0 da rimuovere e 6 non aggiornati.
È necessario scaricare 24,2 MB di archivi.
Dopo quest'operazione, verranno occupati 117 MB di spazio su disco.
Continuare? [S/n] ■
```



Esempi di uso di apt

```
so@so-vbox:~$ apt search openjdk
Ordinamento... Fatto
Ricerca sul testo... Fatto
crypto-policies/focal,focal 20190816git-1 all
    unify the crypto policies used by different applications and libraries

default-jdk/focal 2:1.11-72 amd64
    kit di sviluppo Java standard o compatibile con Java

default-jdk-doc/focal 2:1.11-72 amd64
    kit di sviluppo Java standard o compatibile con Java (documentazione)

default-jdk-headless/focal 2:1.11-72 amd64
    kit di sviluppo Java standard o compatibile con Java (headless)

default-jre/focal 2:1.11-72 amd64
    runtime Java standard o compatibile
```



Documentazione

- La **documentazione** è accessibile dalla shell tramite il comando **man**
man ls
- La documentazione consiste di più sezioni, tra cui
 1. Commands
 2. System calls
 3. Library functions

man 1 ls

man 2 fork

man 3 printf

Premere i tasti freccia per
scorrere il manuale, e il tasto
“q” per uscire