

# L'ambiente UNIX: primi passi

- **Sistemi UNIX multiutente**
  - necessità di autenticarsi al sistema
  - procedura di **login**
- **Login: immissione di credenziali**
  - **username**: nome identificante l'utente in maniera univoca
  - **password**: parola chiave segreta
- **Logout: procedura di uscita dal sistema**
- **SHELL: interprete di comandi**
- **Shell considerata: BASH**
  - Bourne Again Shell (il seguito della shell di Bourne), 1989-2004

# Comandi di base

- **Comandi shell di base (per la sopravvivenza)**
  - **ls**: mostra contenuto di una directory
  - **cd *dir***: entro nella directory *dir*
  - **passwd**: cambio password
  - **file *filename***: stampa il tipo di file di *filename*
  - **cat *filename***: mostra il contenuto di *filename*
  - **pwd**: stampa la directory corrente
  - **exit (logout)**: esegue il logout
  - **man *command***: pagine di manuale su *command*
  - **info *command***: pagine info su *command*
  - **apropos *string***: cerca informazioni su *string* nel database *whatis*

# Opzioni ed argomenti

- Ciascuno dei comandi di shell va battuto in seguito al **prompt**
- Ciascun comando termina con la pressione del testo **Enter**
- I comandi prevedono **opzioni ed argomenti**
  - **opzioni**: specificano la modalità di operazione
  - **argomenti**: specificano su cosa si opera
- **Opzioni**:
  - formato breve ( **ls -a** )
    - ◆ più opzioni con un solo – (ls -al equivalente ad ls -a -l), se non è richiesto un argomento per l'opzione
  - formato esteso ( **ls -all** )

# Combinazioni tasti shell

- **Utili combinazioni tasti bash**
  - **Ctrl+a**: muove il cursore all'inizio della riga
  - **Ctrl+c**: termina un programma in esecuzione
  - **Ctrl+d**: logout
  - **Ctrl+e**: muove il cursore alla fine della riga
  - **Ctrl+l**: pulisce la schermata
  - **Ctrl+r**: cerca la *command history*
  - **Ctrl+z**: sospende un programma
  - **Arrow Up/Down**: naviga la *command history*
  - **Shift+PageUp/Down**: naviga il buffer del terminale
  - **Tab**: completa il comando/nome file
  - **Tab Tab**: mostra i completamenti possibili

# Help in linea: man pages

- **man pages**: pagine di manuale
  - installate con gli applicativi
- **man *sezione comando***
  - **1**: comandi utente
  - **2**: system call
  - **3**: funzioni libreria C
  - **4**: file speciali (/dev)
  - **5**: specifiche formati file (/etc/passwd)
  - **6**: giochi
  - **7**: varie
  - **8**: strumenti amministrazione superutente
  - **9**: documentazione funzioni kernel

# Help in linea: `whatis` e `--help`

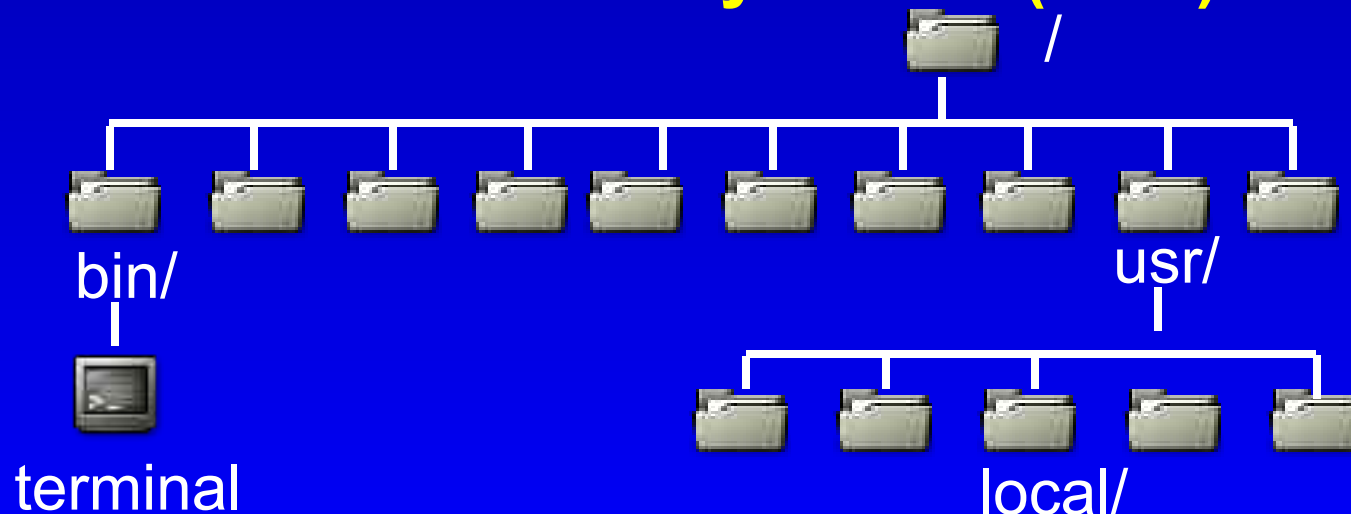
- `apropos`: interroga il database `whatis`
- `apropos stringa`
  - `Es.:` `apropos browser`
- `opzione -help`:
  - mostra brevemente le opzioni di utilizzo di un programma

# Organizzazione dei dati: i file

- In un sistema UNIX, tutto è visibile sotto forma di **file**
  - Ciò che non è visibile sotto forma di file, lo è sotto forma di **processo**
- **File**: contenitore di informazioni (sequenza di byte)
- **Tipi di file**:
  - file regolari
  - directory
  - file speciali
  - link simbolici
  - domain socket
  - named pipe

# Organizzazione gerarchica dei file

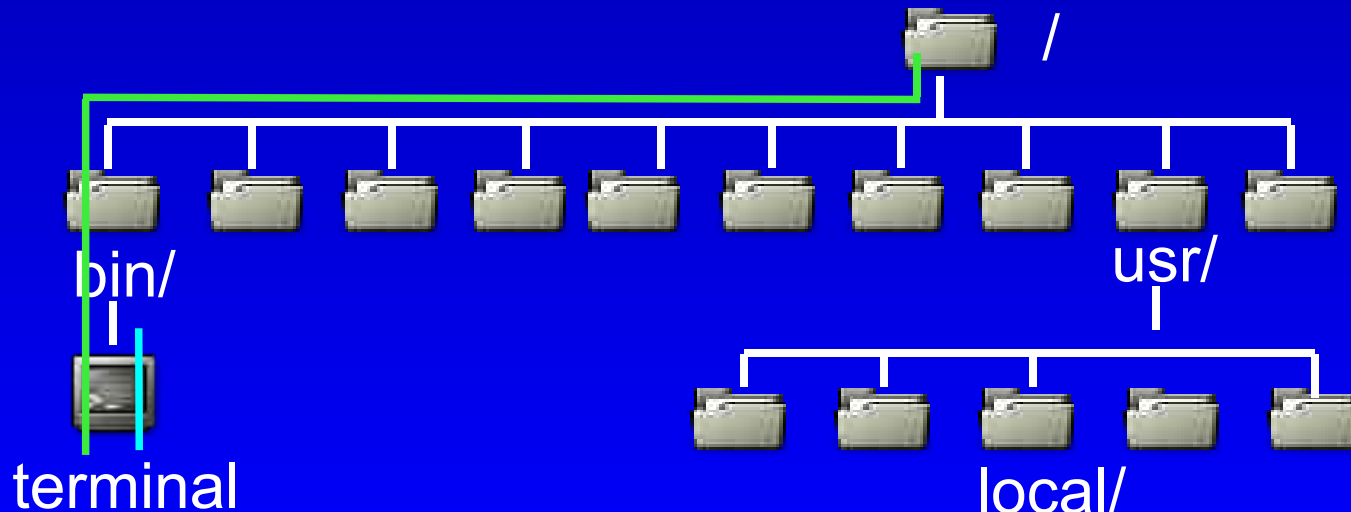
- I file sono organizzati in maniera gerarchica (ad albero)
  - **File:** contenitori di dati
  - **Directory (cartelle):** contenitori di file e di directory
- **Filesystem:** un gruppo di directory e file, a partire da una directory radice (**root**)





# Accesso ai file: percorsi assoluti/relativi

- Ciascun file può essere acceduto seguendo il percorso (path) dalla directory radice al file
  - Path assoluto: si parte dalla root
    - `/bin/terminal` è il path assoluto
  - Path relativo: si parte da un'altra directory
    - `terminal` è il path relativo di `terminal` rispetto alla directory `/bin`



# Accesso ai file: percorsi assoluti/relativi

- **Simboli speciali utilizzati nei percorsi:**
  - **“/”**: indica la directory radice ed è un separatore nel percorso
  - **“.”**: indica la directory in cui l'utente si trova attualmente
  - **“..”**: indica la directory immediatamente superiore a quella attuale
- **Un percorso relativo di solito si indica riferendosi alla directory attuale:**
  - **./sottodir1/sottodir2/file**
  - **./terminal**
- **Come stampiamo la directory corrente?**
  - **comando pwd**

# Link a file

- **Un link può essere visto come una scorciatoia per raggiungere un dato tipo di file**
  - **Se leggo spesso il file `/usr/share/man/man1/lis.1.gz`, devo scrivere il percorso tutte le volte**
- **Gli creo un link**
  - **un file che sta, per esempio, nella mia directory e che “punta” al file in questione**
- **Due tipi di link:**
  - **link fisici (hard link)**
  - **link simbolici (soft link)**

# Link fisici

- **Vengono creati col comando ln**
  - In sorgente destinazione
- **Richiedono permessi di lettura su destinazione**
- **Elemento di directory sorgente puntante al file destinazione**
- **Sorgente e destinazione devono risiedere sulla stessa partizione**
  - poiché il link simbolico è solo una directory entry, in essa non può essere scritta una informazione relativa al disco su cui si trova destinazione
- **Se elencato col comando ls, un link fisico non presenta differenze con l'originale**
- **Per cancellare definitivamente il file in questione, bisogna rimuovere tutti gli hard link con rm**

# Link simbolici

- **Vengono creati col comando ln -s**
  - In -s sorgente destinazione
- **Non richiedono permessi di lettura su destinazione**
- **File speciale (piccolo) contenente un percorso assoluto al file destinazione**
- **Sorgente e destinazione possono risiedere su partizioni diverse**
  - poiché ho il percorso assoluto scritto su file
- **Se elencato col comando ls, un link simbolico presenta differenze con l'originale (sono due file diversi)**
- **La rimozione del link simbolico non comporta la rimozione del file collegato**

# Visione delle directory

- Come stampiamo a video la directory corrente?
  - `pwd`
- Come visualizziamo il contenuto di una directory qualsiasi?
  - `ls directory`
- Alcune opzioni comuni di `ls`:
  - `-a`: visualizza tutti i file, anche quelli nascosti (il cui nome comincia con il carattere `.`)
  - `-r`: visualizza ricorsivamente tutto l'albero di sottodirectory
  - `--color`: output colorato per diversi tipi di file
  - `-l`: formato lungo (tutte le informazioni di un file)

# Utenti e gruppi

- **Gli utenti dei sistemi UNIX sono classificati in user e group**
- **user:** nome identificativo (di solito  $\leq 8$  caratteri) che caratterizza l'utente
  - memorizzati nel file `/etc/passwd` insieme a tutte le loro altre proprietà
- **group:** è un insieme di utenti
  - memorizzati nel file `/etc/group`
- **Ciascun file ha associati:**
  - un utente creatore del file (**owner**)
  - un gruppo di utenti che lavorano sul file (**group**)
- **Utenti e gruppi permettono di garantire permessi di accessi ai file a gruppi di utenti distinti**

# Utenti e gruppi

- **id**: mostra l'identificatore della propria utenza
- Un utente può appartenere a più gruppi, ma può essere attivo in un solo gruppo alla volta
- **newgrp**: permette di passare da un gruppo ad un altro
  - newgrp staff
- **sg**: permette di eseguire comandi sotto un altro gruppo
  - sg <nuovo gruppo> -c <comando>
- **OSS**: anche i gruppi hanno una password!
  - può capitare che newgrp e sg ve la chiedano...



# Permessi dei file

- **Schema UNIX di permessi: rwx**
  - **Read:** permesso di lettura sul file
  - **Write:** permesso di scrittura sul file
  - **eXecute:** permesso di esecuzione del file
- **Tre insiemi di permessi associati ad un file:**
  - **owner:** permessi rwx dell'owner
  - **group:** permessi rwx del gruppo
  - **other:** permessi rwx per tutti gli altri

# Proprietà dei file

Visibili con il comando `ls -l`

```
-rwxr-xr-x 1 root root 75716 Jul 26 17:14 /bin/ls
```

# Proprietà dei file

Visibili con il comando `ls -l`

```
-rwxr-xr-x 1 root root 75716 Jul 26 17:14 /bin/ls
```



- **Tipo di file e permessi particolari**
  - **-**: file regolare (eseguibile, file testo)
  - **b**: dispositivo a blocchi
  - **c**: dispositivo a caratteri
  - **d**: directory
  - **l**: link
  - **p**: fifo
  - **s**: socket

# Proprietà dei file

Visibili con il comando `ls -l`

```
-rwxr-xr-x 1 root root 75716 Jul 26 17:14 /bin/ls
```

- **Permessi dell'owner**
  - **r**: l'owner (root) può leggere il file
  - **w**: l'owner (root) può scrivere il file
  - **x**: l'owner (root) può eseguire il file
- **OSS.:** il bit **x** può anche essere impostato ad
  - **s**: setuid bit (esecuzione con i privilegi dell'owner)
  - **accesso diretto risorse HW da parte di utenti**

# Proprietà dei file

Visibili con il comando `ls -l`

```
-rwxr-xr-x 1 root root 75716 Jul 26 17:14 /bin/ls
```



- **Permessi del gruppo**
  - **r:** tutti gli utenti del gruppo root possono leggere il file
  - **-:** tutti gli utenti del gruppo root non possono scrivere il file
  - **x:** tutti gli utenti del gruppo root possono eseguire il file

# Proprietà dei file

Visibili con il comando `ls -l`

```
-rwxr-xr-x 1 root root 75716 Jul 26 17:14 /bin/ls
```



- **Permessi di tutti gli altri**
  - **r:** tutti gli altri possono leggere il file
  - **-:** tutti gli altri non possono scrivere il file
  - **x:** tutti gli altri possono eseguire il file

# Proprietà dei file

Visibili con il comando `ls -l`

```
-rwxr-xr-x 1 root root 75716 Jul 26 17:14 /bin/ls
```

Numero di hard  
link al file



# Proprietà dei file

Visibili con il comando `ls -l`

```
-rwxr-xr-x 1 root root 75716 Jul 26 17:14 /bin/ls
```

**Owner del file**  
l'utente root

**Gruppo del file**  
il gruppo root



# Proprietà dei file

Visibili con il comando `ls -l`

```
-rwxr-xr-x 1 root root 75716 Jul 26 17:14 /bin/ls
```

↑  
Dimensione del  
file in byte

# Proprietà dei file

Visibili con il comando `ls -l`

```
-rwxr-xr-x 1 root root 75716 Jul 26 17:14 /bin/ls
```

Data ed ora di  
creazione



# Proprietà dei file

Visibili con il comando `ls -l`

```
-rwxr-xr-x 1 root root 75716 Jul 26 17:14 /bin/ls
```

Path assoluto  
del file



# Creazione di file e directory

- **mkdir:** creazione di directory
- **rmdir:** rimozione di directory
- **touch:** creazione di file vuoti
- **rm:** rimozione di file e directory
  - **-r:** cancella in maniera ricorsiva le sottodirectory
  - **-f:** forza la cancellazione senza chiedere
- **OSS.:** in Linux **NON** esiste un meccanismo semplice per l'operazione di undelete (annullamento cancellazione file)
  - se per sbaglio cancellate un file, l'avete perso per sempre
  - a meno che non siate dei bravi hacker...

# Modifica utente e gruppo di un file

- **chown**: permette di modificare owner e gruppo iniziale di un file
  - **chown** <opzioni> <owner>:<gruppo> <file>
  - **chown** <opzioni> <owner>:<gruppo> <directory>
  - **-R**: opera ricorsivamente sulle sottodirectory  
**chown -recursive root:staff /root**

# Modifica permessi di un file

- **chmod**: permette di modificare i permessi associati ad un file
- **Due tipi di sintassi:**
  - **ottale**
    - numeri in codifica ottale associati ai singoli permessi  
r,w,x: r=4, w=2, x=1
    - i permessi sono rappresentati da 3 cifre ottali (755, 644)
  - **testuale**
    - primo set di caratteri rappresenta il tipo di permesso:  
u=user, g=group, o=other, a=all (ugo)
    - il set di caratteri successivi rappresentano una maschera dei permessi per il tipo di permesso selezionato: r=read, w=write, x=execute
    - fra i due set di caratteri viene messo un + per abilitare i permessi oppure un – per disabilitarli
    - posso dare più maschere separate da virgole

# Modifica permessi di un file

- Per dare i seguenti permessi al file prova:
  - owner: rwx
  - group: rwx
  - other: rx

**posso immettere i seguenti due comandi, del tutto equivalenti:**

```
andreoli@romanella ~ $ chmod 755 prova
```

```
andreoli@romanella ~ $ chmod ug+rwx,o+rx prova
```

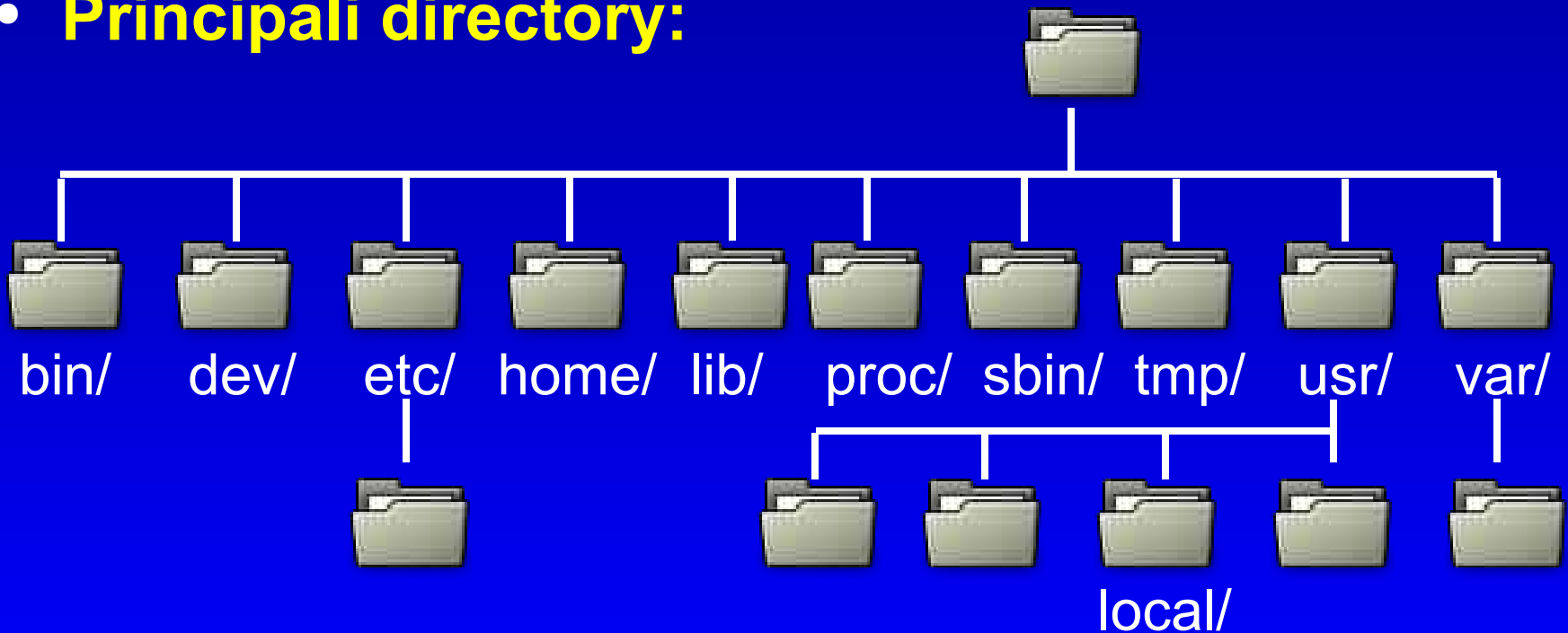
# File speciali

- Localizzati nella directory /dev, servono ad accedere alle risorse del sistema
  - dispositivi a blocchi
    - dischi, cdrom, dvd
  - dispositivi a carattere
    - linee seriali, terminali, mouse
- **Struttura file speciale:**
  - /dev/[device major][device minor][numero]
    - **device major**: nome identificativo della classe di dispositivo (disco EIDE, disco SCSI, )
    - **device minor**: numero identificativo del tipo di dispositivo (disco fisso EIDE, cdrom EIDE)
    - **numero**: identificatore del dispositivo (primo disco, secondo disco, ...)



# Il filesystem / di Linux

- Non esiste il concetto delle “unità” DOS
- Un'unica radice per l'intero filesystem
- Struttura di directory che affonda le radici nella storia pluri-decennale di Unix
- Principali directory:



# Il filesystem / di Linux

- **/bin**
  - Eseguibili principali per far funzionare il sistema
  - Tra le altre cose deve contenere il comando mount
- **/dev**
  - Dispositivi fisici del sistema
  - Filosofia di Unix: tutto è un file
  - Esempi: /dev/hda, /dev/sda, /dev/hda1
  - A partire dalla versione 2.4 di Linux sono state introdotte modifiche importanti per automatizzare la gestione dei dispositivi
    - devfs (kernel 2.4)
    - udev (kernel 2.6)

# Il filesystem / di Linux

- **/etc**
  - Files di configurazione del sistema
  - Struttura di directory potenzialmente molto complessa
  - Contiene anche script per l'avvio del sistema (/etc/init.d)
- **/home**
  - Informazioni personali di ogni utente
  - una sottodirectory per ciascun utente
- **/lib**
  - Librerie di sistema
  - file .so (shared object) equivalenti alle dll di Windows

# Il filesystem / di Linux

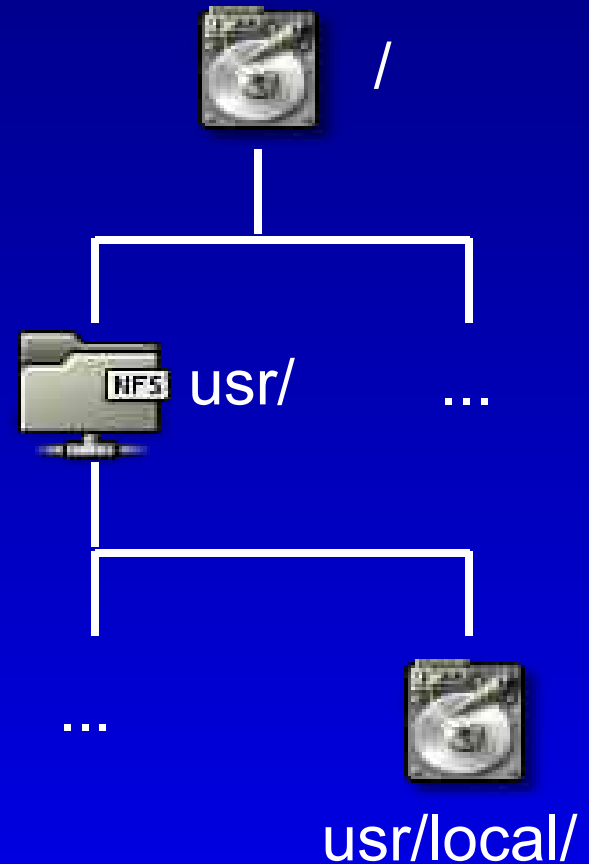
- **/proc**
  - Una finestra sulle strutture dati del kernel
  - Non è una directory che risiede su un disco
  - E' un filesystem montato
  - Ogni processo ha una sua sottodirectory
  - Sono presenti altri file che mostrano/raccolgono informazioni per il kernel
- **/sbin**
  - Contiene eseguibili per il superutente
  - Esempi: shutdown, halt

# Il filesystem / di Linux

- **/tmp**
  - Directory temporanea
  - Tutti hanno diritto di scrittura su tale directory
- **/usr**
  - Albero di directory simile per certi versi alla root directory
  - Qualche tempo fa era usato per trarre vantaggio da dischi condivisi in rete che ospitavano software comune e “voluminoso”
  - Contiene software non essenziale
  - Oggi è ancora molto usato ma rimane per ragioni storiche

# Il filesystem / di Linux

- **Sotto /usr troviamo**
  - /usr/bin, /usr/lib, /usr/sbin che sono analoghi alle controparti nella root directory
- **/usr/local**
  - Software non essenziale per il funzionamento del sistema da non condividere tra più macchine
- **/usr/X11R6**
  - Software legato al sistema grafico X-window



# Il filesystem / di Linux

- **/var**
  - **File che variano spesso**
    - **Log di sistema che vengono scritti molte volte al giorno**
    - **Directory di spool per la posta**
    - **Directory di spool per la stampa**
  - **File temporanei con “vita lunga” (che rimangono anche dopo i reboot) vanno in /var/tmp**

# Filesystem e partizioni

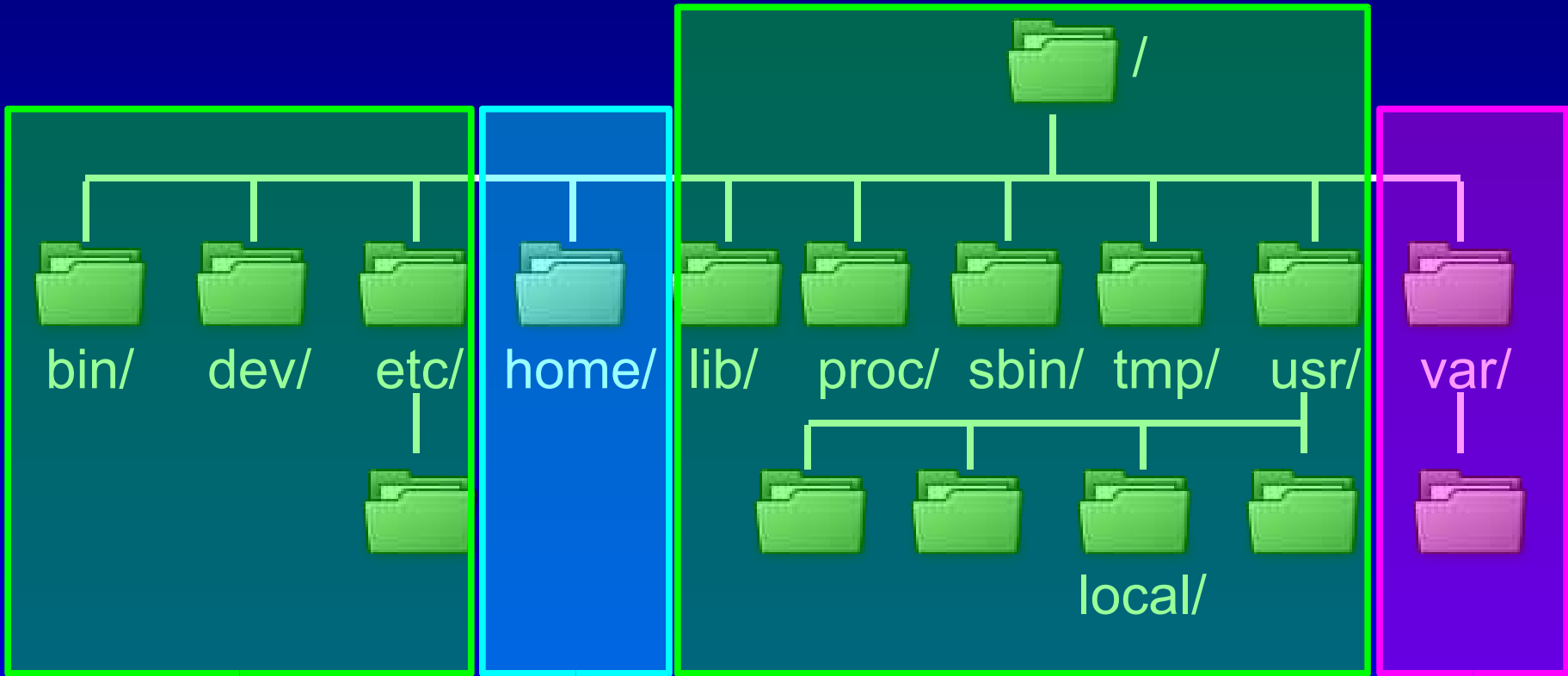
- **Partizionamento dello spazio disponibile su disco in aree logicamente separate**
  - in caso di disastro, si rovina una sola area
  - le altre rimangono intatte
- **Partizioni:**
  - dati (contengono i file)
  - swap (memoria aggiuntiva)
- **Una partizione di dati**
  - è associata ad una directory del sistema, detta **mount point**
  - **contiene un filesystem**
  - **si usa dire:** il filesystem **<fs>** è montato sotto la directory **<dir>**



# Filesystem e partizioni

- **Nomi partizioni:**
  - file speciali (contenuti nella directory `/dev`) identificano il dispositivo
    - ◆ Es.: `/dev/hda`
  - file speciali con attaccato un numero intero indicano la partizione
    - ◆ Es.: `/dev/hda1`, `/dev/hda2`

# Esempio di partizionamento



**Device**

**/dev/hda1**

**/dev/hda2**

**/dev/hda3**

**Mount point**

**/**

**/home**

**/var**

# Gestione delle partizioni

- **fdisk**: gestione partizioni disco
  - creazione, distruzione, visione struttura

romanella ~ # fdisk -l /dev/sda

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1		1	13	104391	de	Dell Utility
/dev/sda2	*	14	1160	9213277+	7	HPFS/NTFS
/dev/sda3		1161	7296	49287420	f	W95 Ext'd (LBA)
/dev/sda5		1161	1288	1028128+	b	W95 FAT32
/dev/sda6		1289	1293	40131	83	Linux
/dev/sda7		1294	1543	2008093	82	Linux swap/Solaris
/dev/sda8		1544	7296	46210941	83	Linux

- **df**: occupazione spazio partizioni montate

andreoli@romanella ~ \$ df

Filesystem	1k-blocks	Used	Available	Use%	Mounted on
/dev/sda8	45484964	14911472	28262948	35%	/

# Tipi di file

- Quali tipi di file riconosce Linux?
  - Multipurpose Internet Mail Extension, RFC2045
  - Standardizzazione dei diversi contenuti di un file
- Linux **NON** adotta la convenzione Microsoft secondo la quale il tipo di file è determinato dalla sua estensione
  - .exe, .com, .doc, .xls non hanno significato
  - l'utente Linux può comunque usare estensioni
  - il SO non le usa per “comprendere” il tipo di file
- Cosa usa il SO per comprendere il tipo di file?
  - una lista di “sequenze magiche” di byte
  - file `/usr/share/misc/files/magic`
  - lunga lista di record del tipo:
    - ♦ offset. tipo dato. valore. tipo di file (MIME)

# Tipi di file

- **file: interfaccia al database dei tipi MIME**
  - **dato un file, vi dice la natura dei suoi contenuti**

```
andreoli@romanella ~ $ file mysong.mp3
```

```
mysong.mp3: MPEG ADTS, layer III, v1, 128 kBits, 44.1 kHz, Stereo
```

```
andreoli@romanella ~ $ file mymovie.avi
```

```
mymovie.avi: RIFF (little-endian) data, AVI, 720 x 576, 25.00 fps,  
video: DivX 5, audio: MPEG-1 layer 3 (stereo, 48000 Hz)
```

```
andreoli@romanella ~ $ file /bin/lis
```

```
/bin/lis: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for  
GNU/Linux 2.4.1, dynamically linked (uses shared libs), stripped
```

```
andreoli@romanella ~ $ file /etc/passwd
```

```
/etc/passwd: ASCII text
```

```
andreoli@romanella ~ $ file /dev/sda
```

```
/dev/sda: block special (8/0)
```

# Copia e spostamento di file e directory

- **cp: copia file/directory**
  - **cp** <opzioni> <sorgente> <destinazione>
  - **-r**: copia ricorsivamente sorgente in destinazione (con tutte le sottodirectory)
  - **-p**: preserva (quando possibile) utenze e permessi originali dei file
  - **-a**: modalità archivio (equivalente a -r -p, ed altro)
- **mv: sposta file/directory**
  - **mv** <opzioni> <sorgente> <destinazione>

# Ricerca di file

- **find**: cerca il file richiesto in un albero di directory
  - **find <directory> -name <nome file>**
  - **find <directory> -iname <nome file>**
  - **find <directory> -name <nome file> -exec <comando> {} \;**
  - **lento (si scorre tutte le sottodirectory)**
- **locate**: interroga un database di locazioni dei file
  - **aggiornato tramite il comando **updatedb****
  - **veloce**
  - **non permette di eseguire comandi sui risultati**

# Ricerca di contenuti nei file

- **grep**: cerca la stringa richiesta nell'albero di directory specificato, e stampa le corrispondenze (**match**)
  - **grep <opzioni> <stringa> <directory/file>**
  - **-r**: opera ricorsivamente nelle sottodirectory
  - **-i**: trascura maiuscole/minuscole (case insensitive)
  - **-H**: stampa il file contenente la stringa
  - **-n**: stampa il numero di riga del file contenente la stringa



# Visione dei contenuti di un file

- **cat:** stampa il contenuto del file su terminale
- **more:** visione di un file non bufferizzata (solo in avanti)
- **less:** visione di un file bufferizzata (avanti e indietro)
  - **less -Mr** presenta un indicatore di progresso di lettura
- **head:** stampa le prime n righe di un file
  - **head -n 10 file**
- **tail:** stampa le ultime n righe di un file
  - **tail -n 10 file**
  - **tail -f file** output bufferizzato
- **echo:** stampa una stringa su terminale

# Identificazione di file: espressioni regolari

- **less:** visione di un file bufferizzata (avanti e indietro)
  - `less -Mr` presenta un indicatore di progresso di lettura
- **head:** stampa le prime n righe di un file
  - `head -n 10 file`
- **tail:** stampa le ultime n righe di un file
  - `tail -n 10 file`
  - `tail -f file` output bufferizzato
- **echo:** stampa una stringa su terminale