



SAPIENZA
UNIVERSITÀ DI ROMA

**Corso di laurea in Ingegneria
dell'Informazione
Indirizzo Informatica**

Reti e sistemi operativi

**Introduzione a Linux:
nozioni base e comandi
fondamentali.**

Introduzione (1/2)

- GNU/Linux è di base un kernel multitasking e multiutente ma ...
- ... con Linux ormai si identificano tutti quei sistemi operativi Unix-like basati sull'omonimo kernel che integrano, tra gli altri, componenti GNU/GPL.
- Perché GNU? (utility, librerie C, compilatore, ...) e perché Linux? (Unix, Minix, Linus Torvalds,...).

Introduzione (2/2)

- Linux è un progetto libero e aperto ai (fondamentali) contributi esterni.
- Esistono centinaia di distribuzioni ("distro"): sono sistemi operativi pronti all'uso, basati sul kernel Linux, che comprendono utility e suite di applicazioni.
- (K/X-)Ubuntu è la distribuzione desktop attualmente di maggior successo, grazie ad una semplicità inedita per il mondo Linux.

La shell (1/2)

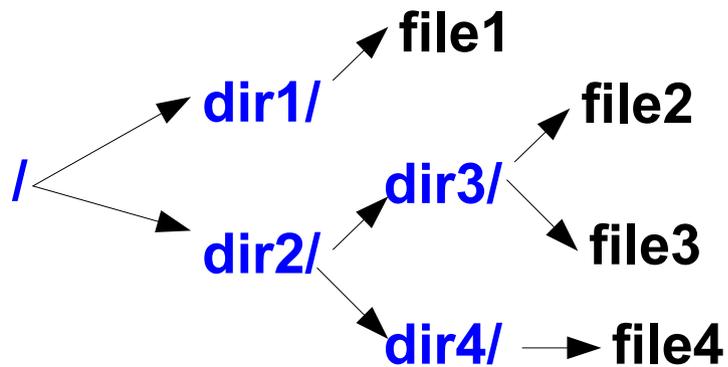
- Ovvero: (quasi) tutto a portata di dita!
- E' un programma che permette agli utenti di comunicare con il sistema attraverso una serie di comandi e di avviare altri programmi da linea di comando (CLI, Command Line Interface).
- In Linux: bash (Bourne again shell).
- Anche Microsoft ha riconosciuto l'importanza della linea di comando proponendo Windows Powershell.

La shell (2/2)

- Nei sistemi embedded Linux (e non solo) l'interfaccia grafica non è (quasi) mai disponibile.
- E' però sempre disponibile la shell, a cui possiamo accedere via tastiera e monitor, via seriale (in windows COM1, ecc... , in linux /dev/ttySx) o via rete (es. Ethernet).
- La shell (bash) in Kubuntu: Konsole

Accedere al filesystem (1/5)

- Filesystem: *come* sono memorizzati i dati in un dispositivo (hard-disk, dvd, ecc...).
- Esistono diversi tipi di file system (NTFS, ext4, ecc...) ma tutti organizzano il contenuto dei dischi con una struttura ad albero con radice / (root), nodi (directory) e foglie (file).



Esempio: path (percorso) assoluto a file3:

`/dir2/dir3/file3`

- **File** è un **contenitore di informazione digitalizzata**, ma non solo...
- In Linux *all is a file* (almost!)

Accedere al filesystem (2/5)

- Il prompt: **pinco_pallo@my-laptop:~\$**_

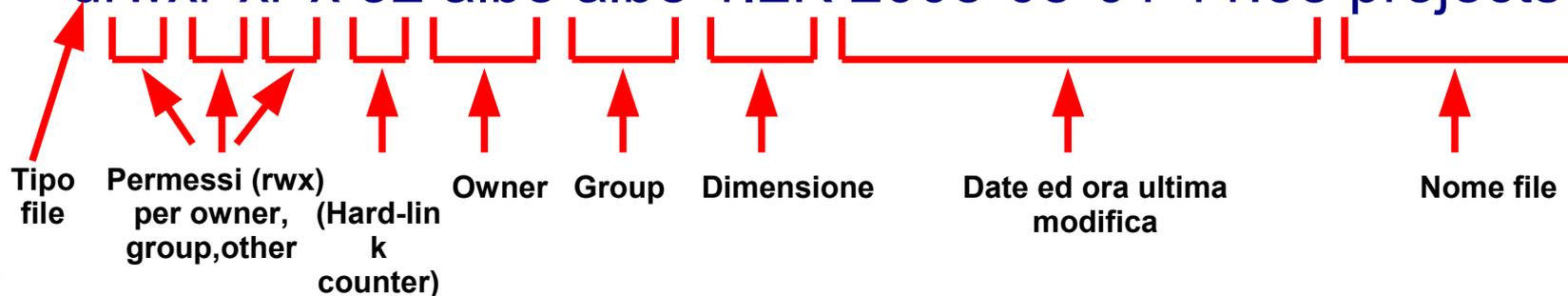
- Dove sono? **pwd**

- Cosa contiene la directory corrente? **ls**

esempi: **ls -lSh**, **ls -a**, **ls -t**, **ls -R**

```
-rw-r--r--    1 albe albe 401K 2008-10-16 16:25 robots.jpg
```

```
drwxr-xr-x   32 albe albe 1.2K 2008-08-01 11:38 projects
```



- Sembra non molto intuitivo! Invece ...

List = **ls**, **ls --help**, **man ls** (provare a cercare un termine all'interno della documentazione (tasto /), **q** per uscire dal **man**)

Accedere al filesystem (3/5)

- Cambiare directory: **cd**, Es: **cd dir**, **cd ..**, **cd /**
(**..** e **.** = file speciali che rappresentano la directory padre e la directory corrente, rispettivamente)
- Creare una directory vuota: **mkdir** <dirname>
- Creare un file vuoto: **touch** <filename>
- Cancellare un file: **rm** <filename>
- Cancellare una directory: **rm -r** <dirname>
- Copiare un file o directory: **cp [-r]** <source> <dest>
- Spostare (e rinominare): **mv** <source> <dest>

Accedere al filesystem (4/5)

- Cambiare i permessi, il gruppo o il proprietario di un file:
 - `chmod XYZ <filename>`
 - `chown <owner> <file>`
 - `chgrp <group> <file>`
- **ATTENZIONE:** I file e le directory che iniziano con `.` sono invisibili all'utente (per visualizzarli **ls -a**)

Accedere al filesystem (5/5)

- Creare un collegamento: **ln -s <file> <linkname>**
In Linux un collegamento così creato viene chiamato soft, ovvero simbolico. Funziona sia per i file che per le directory.
- I collegamenti sono trasparenti ai comandi.
- **Esercizio:** creare un file, creare una directory, creare all'interno della directory un collegamento al file creato, verificare il collegamento con **ls <...>**, creare un file nascosto, verificare con **ls <...>**, cancellare tutto.

Directory fondamentali (1/2)

- **cd / && ls** (&& = esegue il comando a destra dell'operatore se il comando a sinistra è stato eseguito senza errori, con || il comando a destra viene eseguito comunque)
- /bin: Programmi essenziali al funzionamento della macchina utilizzabili da tutti gli utenti.
- /sbin: Programmi essenziali al funzionamento della macchina utilizzabili dall'amministratore (utente **root**).
- /etc : File di configurazione.
- /tmp : Directory per i file temporanei.
- /boot : Directory che contiene i file di avvio della macchina (e.g., il **kernel!**).
- /home : Contiene le directory degli utenti. Per l'utente root è /root.

Directory fondamentali (2/2)

- /usr: contiene gli applicativi installati dall'utente (es. Firefox, matlab, ecc...) → comando **whereis** ls
- /lib : Contiene le librerie principali del sistema.
- /var : Directory in cui i programmi archiviano i file temporanei, i file di lock e i log di sistema.
- /dev : (**filesystem di solito virtuale**) Contiene i file speciali che rappresentano i dispositivi hardware (reali e virtuali) connessi alla macchina. Ogni file rappresenta un dispositivo (*all is a file!*), e contiene le informazioni per interagire con quel determinato dispositivo.
- /proc: (**filesystem virtuale**) Contiene i file speciali che rappresentano i processi presenti sul sistema con i loro relativi stati.
- /sys: (**filesystem virtuale**) Contiene i file speciali che riportano le informazioni relative ai dispositivi connessi alla macchina.

Manipolazione dei file (1/2)

- Concatenare il contenuto di una lista di file:
cat file1 file2 ...
- Visualizzare diviso in pagine il contenuto di un file di testo:
less <filename>
- Editare una file:??? (es. **nano**, **vim**, **emacs**, ...)
- Cercare all'interno di un file le righe che contengono un determinato pattern: **grep** [opzioni] <pattern> [file1 file 2...]
Esempio: **grep --color -i -r "l.*x" ***
wildcard "." : uno, e un solo, carattere
wildcard ".*" : 0 o più caratteri
....

Manipolazione dei file (2/2)

- Trovare un file a partire dalla directory attuale:

find <percorso> [-name/-iname] <pattern>

Esempio: `find -iname "*.pdf" .`

wildcard "?": uno, e un solo, carattere

wildcard "*": 0 o più caratteri

...

- Confrontare il contenuto di due file:
diff file1 file2

I processi (1/2)

- **ps** : processes snapshot (lista processi dell'utente associati al terminale corrente)
- **ps aux**
a= processi non solo di questo utente, u= formato visibile all'utente, x=processi anche non associati a terminale.
- Killare un processo: **kill -<segnale> pid**
segnali di esempio: 15 (TERM, può essere intercettato) 9 (KILL, non può essere intercettato).
- Monitorare in tempo reale i processi attivi e la loro occupazione di risorse: **top**

I processi (2/2)

- Campi dell'output di **ps** e **top**:
 - **PID** : l'ID (univoco) del processo
 - **USER** : il proprietario del processo (i.e., chi l'ha lanciato)
 - **PR** : priorità del processo
 - **%CPU** : percentuale di occupazione della CPU (tempo)
 - **%MEM** : percentuale di utilizzo della memoria fisica (spazio)
 - **VSZ/VIRT** : dimensione della memoria virtuale occupata dal processo
 - **RSS/RES** : dimensione della memoria fisica occupata dal processo
 - **SHR** : dimensione della memoria condivisa occupata dal processo
 - **STAT/S** : stato del processo
 - **START** : istante in cui il comando è stato eseguito
 - **TIME** : tempo totale di utilizzo della CPU
 - **COMMAND** : programma/comando che ha generato il processo

Redirezione dell'output

- Redirezione su file dell'output di un comando: >
esempio: **echo** "linux" > file.txt
(**echo** scrive in output la stringa passata come parametro)
Attenzione: se esiste già un file di nome file.txt, il contenuto verrà sovrascritto! Invece:
>> appende (provare!)
- Redirezione dell'output per usarlo come input di un altro comando: | (pipe)
Esempio: **ps** aux | **grep** firefox

Networking - basi (1/2)

- Prima di tutto... **sudo** ... per diventare amministratore! :-)
- Mostra le interfacce di rete presenti nella macchina: **ifconfig**
lo: interfaccia di loopback
ethX: interfaccia ethernet
wlanX: interfaccia wifi
- Configura una interfaccia di rete:
sudo ifconfig <interfaccia> <ip> up

Networking - basi (2/2)

- Verifica la presenza di un host all'interno di una rete: **ping** <ip>
- Connettiti ed esegui comandi shell in una macchina remota provvista di server Secure Shell: **ssh** utente@indirizzo
- Copia un file locale su una macchina remota provvista di server Secure Shell:
scp <file_locale> utente@indirizzo:

Bash tricks

- Scorrere l'output: **Shift + (Up or Down or Pag Up or Pag Down)**
- Completamento automatico (sia per i comandi che per gli argomenti, se essi sono dei file):
tasto **Tab**
- Richiamare gli ultimi comandi eseguiti: tasti **Up/Down**
- Accedere alla history: comando **history**
Eseguire un comando in history: **!**<numero>****