

# Introduzione ai sistemi UNIX/Linux

Corso di laurea magistrale in Ingegneria Elettronica

Ing. Simone Brienza

E-mail: [simone.brienza@for.unipi.it](mailto:simone.brienza@for.unipi.it)

Pervasive Computing & Networking Lab (PerLab) <http://www.perlab.it>

Dipartimento di Ingegneria dell'Informazione, Università di Pisa

# Informazioni generali

---



- Pagina web del corso  
<http://vecchio.iet.unipi.it/se>

# Sommario

---



- Introduzione
  - informazioni storiche
  - caratteristiche
- Installazione di Linux
  - operazioni preliminari
  - installazione
- Elementi fondamentali
  - organizzazione del filesystem
  - shell
  - comandi base
  - altri comandi

# Breve storia di Unix

---



- **1969** - Prima edizione di UNIX sviluppata da Ken Thompson e Dennis Ritchie presso i Bell Laboratories.
- **1973** - Thompson e Ritchie riscrivono UNIX in C.
- Derivati di Unix:
  - BSD (Berkeley Software Distribution).

# Progetto GNU, FSF, GPL



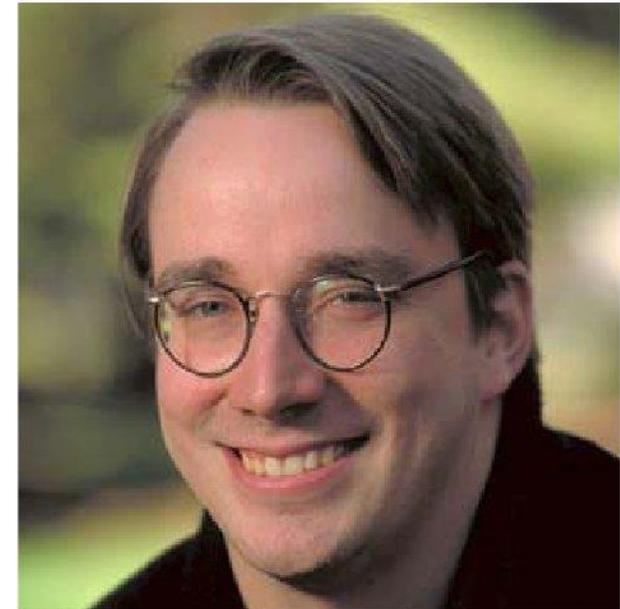
- **1983**, Stallman lancia il progetto GNU con lo scopo di sviluppare un sistema operativo Unix-like libero.
- **1985**, Stallman fonda la Free Software Foundation (FSF), che promuove lo sviluppo di software libero ed, in particolare, del progetto GNU.
- **1989**, Stallman e Moglen scrivono la versione 1.0 della GPL (o GNU GPL, GNU General Public License).
- La GPL è una licenza per software libero.



# GNU/Linux



- GNU/Linux è un sistema operativo Unix-like costituito dall'integrazione del kernel Linux con elementi del sistema GNU.
- Prima versione rilasciata con licenza GPL.
- Esistono numerose varianti (distribuzioni) tra cui:
  - Debian
  - Fedora
  - Ubuntu



# Caratteristiche



- **Caratteristiche fondamentali**
  - **Multitasking:** è possibile la contemporanea esecuzione di più processi a divisione di tempo.
  - **Multi utenza:** più utenti (con diversi privilegi) possono interagire contemporaneamente.
  - **Portabilità:** grazie all'utilizzo del linguaggio C nella realizzazione del sistema.
  - **Modularità:** suddivisione in moduli.

# Struttura di Unix



- 2 componenti principali:
  - kernel: il cui scopo è interagire con l'hardware.
  - applicazioni: che si rivolgono al nucleo per ottenere i servizi richiesti dalle loro funzioni:
    - ⇒ interpreti dei comandi (shell).
    - ⇒ programmi di sistema (strumenti messi a disposizione dal s.o.)
    - ⇒ programmi utente.

# Installazione di Linux

# Modalità di installazione



- Ci sono 2 possibilità per installare un sistema operativo:
  - ✓ Partizionare l'hard disk e sfruttare il multi-boot
    - **Si riduce lo spazio sull'HD dedicato ad un SO e lo si assegna ad un altro**
    - è un'operazione delicata: un errore in fase di partizionamento può compromettere il funzionamento del computer
    - È possibile installare solo un numero limitato di sistemi operativi
  - ✓ Utilizzare una macchina virtuale
    - **Si esegue un sistema operativo come fosse un'applicazione di un sistema operativo pre-esistente**
    - È una soluzione meno efficiente rispetto alla precedente (le risorse del computer sono condivise fra i due SO)
    - Semplice creazione e gestione delle Virtual Machines (VM)
    - È possibile creare una moltitudine di VMs

# Partizionamento (1/3)



## Preparazione del disco

- Operazioni preliminari
  - fare spazio su disco
- Ridimensionamento partizione Windows
  - direttamente in fase di installazione del S.O.
  - prima di procedere con l'installazione
    - ⇒ Partition Magic
    - ⇒ gparted (<http://gparted.sourceforge.net>)
- consigliata deframmentazione preventiva

# Partizionamento (2/3)



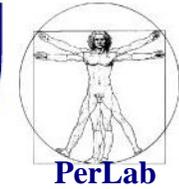
- Partizioni primarie
  - specificate nella tabella delle partizioni del **Master Boot Record** (MBR)
  - al più quattro partizioni
  - una sola avviabile
  - il **Boot Loader** (i.e., GRUB) carica il sistema operativo dalla partizione su cui è stato installato
    - Tipicamente, all'avvio del PC chiede all'utente quale SO avviare
- Partizioni estese
  - permettono di avere delle sotto-partizioni (partizioni logiche)
  - si può così superare il limite delle quattro partizioni primarie)



## Partizioni Linux

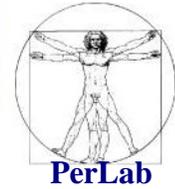
- Per Linux in genere si usano almeno due partizioni per
  - la root (/) per il sistema operativo
  - lo spazio di swap per la memoria virtuale
- Si possono anche utilizzare altre partizioni per
  - la home (/home) per le informazioni degli utenti
  - la cartella `/etc` per i file di configurazione
  - la cartella `/boot` per i file di avvio

# Virtualizzazione



- Si esegue un SO come fosse un'applicazione di un altro preesistente
  - Il sistema operativo ospitante è detto **Host**
  - Il sistema operativo ospitato è detto **Guest**
  - È necessario stabilire quante/quali risorse sono assegnate al SO Guest  
(le risorse del computer sono condivise fra host e guest)
  
- La virtualizzazione è possibile grazie ad alcuni software:
  - VMWare Player  
[https://my.vmware.com/web/vmware/free#desktop\\_end\\_user\\_computing/vmware\\_player/7\\_0](https://my.vmware.com/web/vmware/free#desktop_end_user_computing/vmware_player/7_0)
  - VirtualBox  
<https://www.virtualbox.org/>

# Installazione



- Dopo aver effettuato il partizionamento o aver creato la macchina virtuale, è necessario procedere con l'installazione
- Per installare **Ubuntu** (versione 14.04.4 LTS o seguenti) è possibile scaricare il file immagine dal sito <http://www.ubuntu.com/download/desktop>
- Una volta scaricata l'immagine ISO
  - In caso di partizionamento, la ISO dovrà essere copiata su un CD o una penna USB, da cui sarà necessario effettuare il boot
  - Per creare una Virtual Machine, si può invece sfruttare direttamente la ISO
- Una volta avviata l'installazione, si seguano le istruzioni a video

# Elementi fondamentali

# Creazione degli utenti



- Utente root
  - superutente, amministratore del sistema
  - può compiere qualsiasi tipo di operazione
- Utente normale
  - utilizzatore del sistema
  - ha dei privilegi limitati
- Fase di creazione degli utenti
  - l'utente root deve essere sempre creato
  - conviene creare un account utente normale per l'utilizzo abituale del sistema operativo

# Introduzione al filesystem

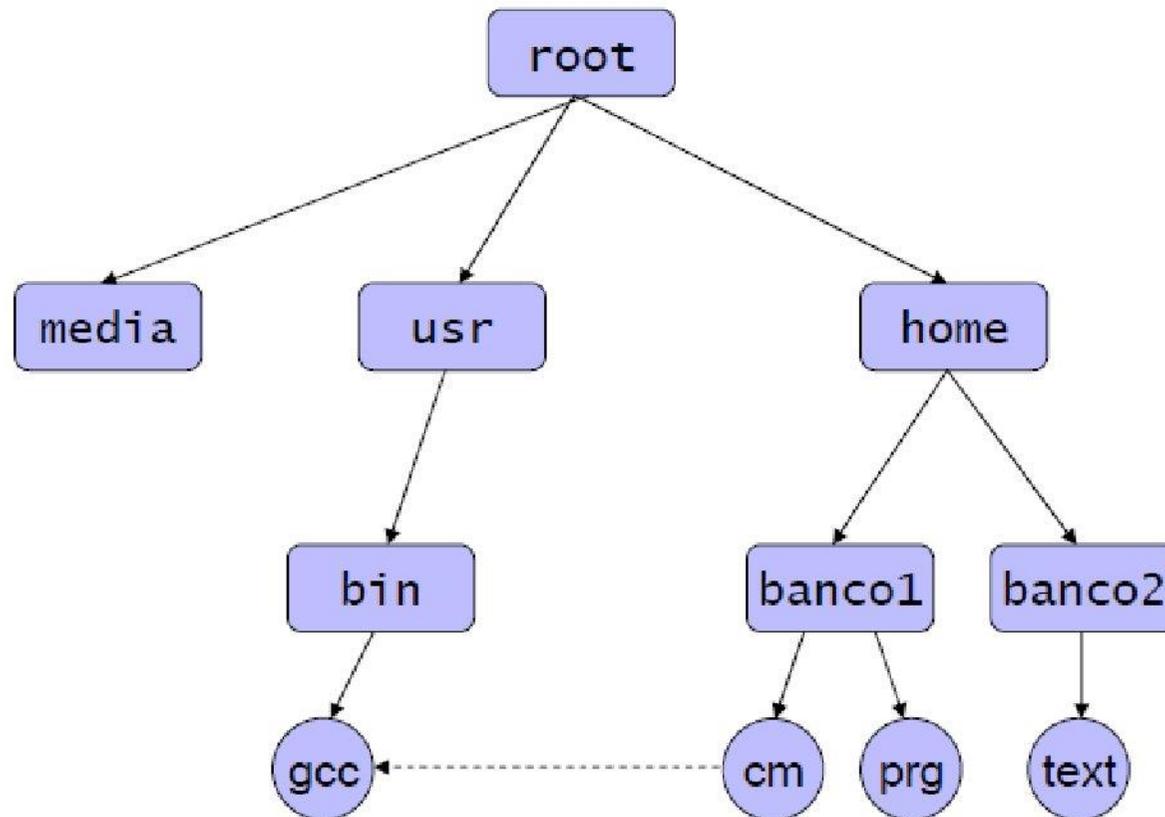


- Necessità di memorizzazione dei dati
- File
  - unità logica di memorizzazione dei dati
  - insieme di informazioni accessibili attraverso il nome del file stesso
- Cartella o directory
  - insieme di file e cartelle
- Filesystem
  - sistema di immagazzinamento e di organizzazione dei file

# Organizzazione dei filesystem



- Struttura a grafo orientato

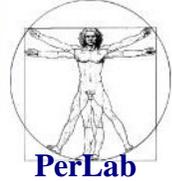


# Organizzazione del filesystem UNIX



- Sottocartelle della cartella principale / (root)
  - `/bin` programmi eseguibili dall'utente
  - `/dev` file speciali corrispondenti ai dispositivi
  - `/etc` file di configurazione del sistema
  - `/home` cartelle personali degli utenti
  - `/lib` librerie condivise
  - `/opt` package che non richiedono installazione
  - `/tmp` file temporanei
  - `/usr` dati condivisi in sola lettura
  - `/var` file di dimensione variabile (es. file di log)

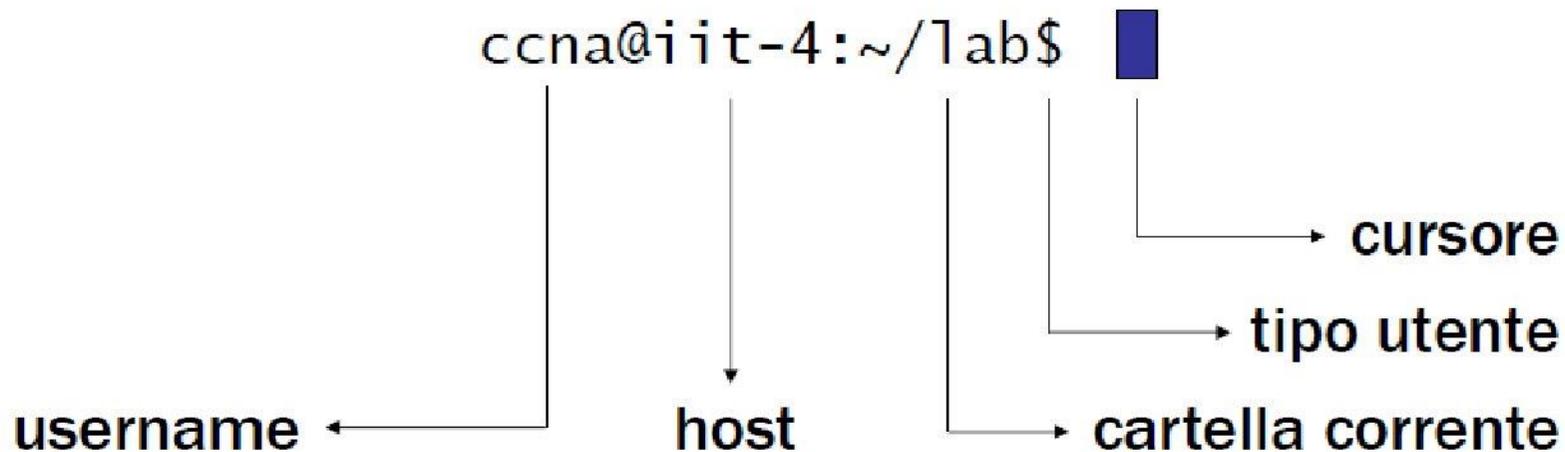
# Path



- *Percorso assoluto*: si esprime l'intero percorso partendo da root (/)
- *Percorso relativo*: si esprime partendo dalla directory di lavoro
  - . directory corrente
  - .. directory padre



- Permette all'utente di interagire con il S.O. mediante l'inserimento di comandi su terminale
- Raccoglie l'input dell'utente e mostra il corrispondente output
- Prompt dei comandi





- Funzioni
  - autocompletamento (tasto TAB)
  - history (freccia SU/GIU)
- Attenzione
  - i filesystem UNIX-like sono case sensitive
  - maiuscole e minuscole sono importanti
  - esempio
    - ⇒ file1, File1, FILE1, FiLe1, sono tutti file diversi

# Accesso al sistema

---



- Login:
  - *login*: nome utente
  - *password*: password
- Disconnessione:
  - *logout* oppure **Ctrl-D**

# Shutdown

---



- I sistemi UNIX permettono solo all'utente root di avviare la procedura di arresto del sistema con i comandi seguenti:
  - `# shutdown -p now`
  - `# shutdown -r now` (riavvio finale).

# Comandi di base

# Comando cd

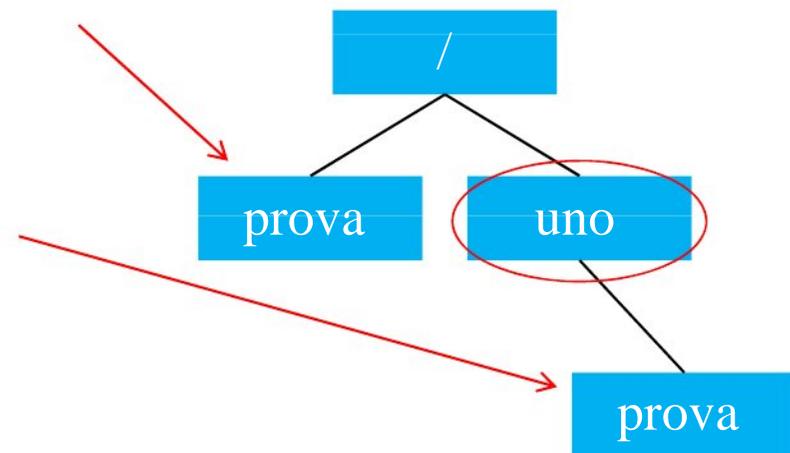


- `cd` (change directory) consente di passare da una directory ad un'altra.
- La sintassi generale di `cd` è la seguente:

`cd [directory]`

Esempio dove sia uno la directory corrente, se si digita:

- `$ cd /prova` (path assoluto)
  - ⇒ si passa alla directory prova;
- `$ cd prova` (path relativo)
  - ⇒ si passa alla directory prova;
- `$ cd ~` o `cd`
  - ⇒ Si passa alla directory home dell'utente;
- `$ cd ..`
  - ⇒ Si passa alla directory padre che, in tal caso, è /.



# Comando pwd



- `pwd` (print working directory) visualizza il percorso **assoluto** della directory corrente.
- Esempio:
  - `$ cd /bin`
  - `$ pwd`
  - `/bin`



- **ls** (list segments) visualizza i nomi di file o il contenuto delle directory indicate.
- La sintassi generale di ls è la seguente:  
`ls [opzioni] [file/dir1 file/dir2...]`
- `[file/dir1 file/dir2...]` : specificano i file e/o le directory da elencare, in loro assenza è elencato il contenuto della directory corrente.
- `[opzioni]` : definiscono le opzioni con cui eseguire il comando ls. Tali opzioni vengono specificate dopo il simbolo - .



## Opzioni principali:

- `$ ls -l`
  - Visualizza, oltre ai nomi, altre informazioni quali ad esempio modalità di accesso, proprietario del file,...
- `$ ls -a`
  - vengono elencati anche gli elementi i cui nomi iniziano con punto (i cosiddetti file nascosti).
- Le opzioni sono “cumulabili”, ad esempio è possibile scrivere:
  - `$ ls -la`



Esempio:

- per vedere il contenuto della directory / (1):

```
$ cd /
```

```
$ ls
```

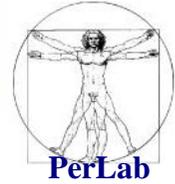
```
bin dev home mnt tmp boot etc lib root  
var
```

- per vedere il contenuto della directory / (2):

```
$ ls /
```

```
bin dev home mnt tmp boot etc lib root  
var
```

# Wildcard



- Insiemi di file possono essere riferiti usando i metacaratteri (wildcard).
  - \*: sostituisce zero o più caratteri.
  - ?: sostituisce un carattere singolo.
  - [a,b,c] o [a-z]: sostituisce un carattere nell'insieme.
- Esempio:
  - `ls *.c`
  - Elenca i file che terminano con `.c`



- Documentazione consultabile tramite il comando `man` (manual).
  - suddiviso in sezioni in base al tipo di argomento trattato.
- La sintassi generale è la seguente:  
`man [n] nome_comando`  
⇒ `[n]` numero di sezione
- Esempio:
  - `$ man 1 printf`



Altri due comandi di help disponibili in Unix sono:

- **whatis**: fornisce la riga sommario relativa ad un comando.
- **apropos**: ricerca per parola chiave (opera in modo analogo al comando `man -k`).

Altri comandi  
Comandi sui file

# Comandi mkdir, rmdir, cp, mv



- `mkdir [opzioni] directory...`: crea una directory.
- `rmdir [opzioni] directory...`: elimina le directory specificate solo se sono vuote.
- `cp [opzioni] origine destinazione`: copia file o directory in un'unica destinazione.
- `mv [opzioni] origine destinazione`: sposta i file e le directory.

# Comandi touch, cat, rm



- `touch [opzioni] file...` : cambia data e ora di accesso e di aggiornamento dei file. Se si specificano file che non esistono, questi vengono creati vuoti.
- `cat [opzioni] file...`: concatena dei file e ne emette il contenuto attraverso lo standard output.
- `rm [opzioni] file...`: Rimuove i file indicati come argomento. In mancanza dell'indicazione delle opzioni necessarie, non vengono rimosse le directory.

# Comandi lettura da file

---



- `less [opzioni] [file]...`
- `more [opzioni] [file]...`
- Utilizzati per leggere file di puro testo.

# Operazioni sui file

---



- `tail [opzioni] [file]... :`  
mostra le ultime linee di dati provenienti da uno o più file di testo.
- `head [opzioni] [file]... :`  
mostra le prime linee di dati provenienti da uno o più file di testo.

# Redirezione e pipeline



- La redirezione reindirige i dati in modo da destinarli ad un file o da prelevarli da un file (uso di `<`, `2>`, `>`, `>>`).
- Redirezione dell'input (`<`)
  - `$ cat < elenco`
    - ⇒ Si ottiene in questo modo la visualizzazione del contenuto del file elenco.
- Redirezione dello standard error (`2>`)
  - `$ ll 2> elenco`
    - ⇒ Salva lo standard error nel file elenco.



- Redirezione dell'output (>, >>)
  - `$ ls -l > elenco`
    - ⇒ Questo comando genera il file elenco con il risultato dell'esecuzione di ls.
  - Genera un nuovo file ogni volta, eventualmente sovrascrivendo ciò che esiste già con lo stesso nome.
  - La redirezione dell'output può essere fatta in aggiunta, aggiungendo i dati ad un file esistente con l'uso di >>:
    - `$ ls -l /tmp >> elenco`
      - ⇒ In tal modo viene aggiunto al file elenco l'elenco dettagliato del contenuto della directory /tmp.

# Pipeline

---



La pipeline è una forma di redirectione in cui la shell invia l'output di un comando come input del successivo.

- `$ cat elenco | sort`
  - In questo modo, `cat` legge il contenuto del file `elenco` che, invece di essere visualizzato sullo schermo, viene inviato dalla shell come input di `sort` che lo riordina e poi lo emette sullo schermo.

# Risorse e riferimenti



- Riferimenti su dispensa:
  - 'Amministrazione di un Sistema UNIX in Rete', Cap 1 e 2
- Altre risorse
  - Informazioni specifiche su Ubuntu:
    - ⇒ "Ubuntu Pocket Guide and Reference"  
<http://www.ubuntupocketguide.com/>
  - Partizionamento del disco
    - ⇒ GParted Live  
<http://gparted.sourceforge.net/livecd.php>
    - ⇒ Parted Magic  
<http://partedmagic.com/>
  - Virtual machine free:
    - ⇒ VMware Player