

Breve storia su 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**, Stallmann 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

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.

Caratteristiche di Unix



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

Partizionamento

Un hard disk può essere partizionato in più partizioni:

- **partizioni *primarie***
 - specificate nella tabella delle partizioni del master boot record (MBR)
 - al più 4 partizioni primarie in un hard disk
 - una sola avviabile
- **partizioni *estese***
 - partizioni primarie partizionate in sotto-partizioni
 - (partizioni logiche)
 - permettono di superare il limite delle quattro partizioni primarie

Partizionamento

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

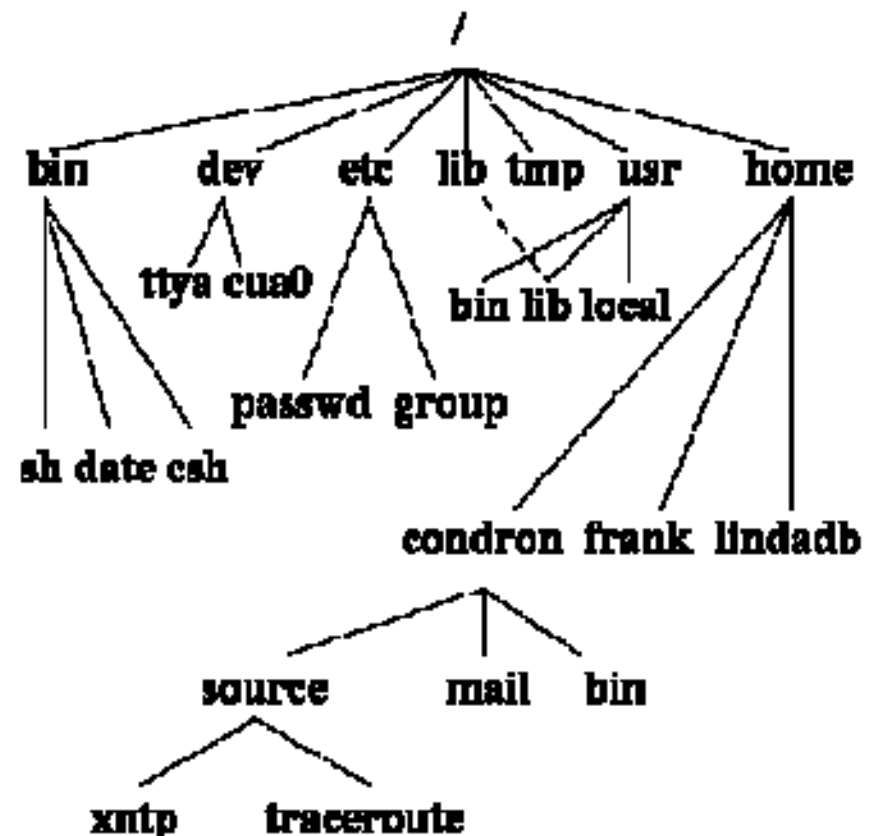
File e Directory

- *File*: sequenza di 0 o più byte contenenti informazioni il cui significato è stabilito dal proprietario del file.
- *Directory*: specifica entità del *file system* che elenca altre entità, tipicamente *file* e altre *directory*.

File system



- Struttura ad albero.
- Ogni nodo è o un file o una directory di file.



Cartelle predefinite

- Sottocartelle della cartella principale / (root):
- /bin programmi eseguibili dall'utente
- /dev file dei device
- /etc file di configurazione del sistema
- /lib librerie condivise
- /tmp file temporanei
- /usr comandi e strumenti dell'utente
- /root home dell'utente root
- /home cartelle personali degli utenti

Path

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

Shell

- Esistono varie shell (sh, bash, csh, tcsh)
- La shell è un interprete di comandi che ripetutamente:
 - stampa un prompt
 - legge un comando scritto dall'utente e terminato con enter/return/invio
 - esegue il comando o segnala un errore se non è in grado di completarlo

Interprete di comandi

osor@wind4:~/lab\$

username ← host ← cartella corrente ← cursore

- UNIX è case-sensitive

Utenti

- **Utente root**
 - amministratore del sistema,
 - può compiere qualsiasi tipo di operazione.
- **Utente normale**
 - utilizzatore del sistema,
 - ha dei privilegi limitati.
- conviene creare un account utente normale per l'utilizzo abituale del sistema operativo.

Accesso al sistema

- **Login:**
 - *login:* nome_utente
 - *password:* password
- **Disconnessione:**
 - `logout` oppure **Ctrl-D**
- **Funzioni**
 - autocompletamento (tasto **TAB**)
 - history (freccia **SU/GIU**)
- **Terminali Virtuali:**
 - **Alt + F1 (F2,F3,...)** : passaggio ad altri terminali.

Shutdown

- I sistemi UNIX permettono solo all'utente root di avviare la procedura di arresto del sistema con i comandi seguenti:
 - # shutdown -h now
 - # halt

Comandi di Base

Comando cd

- `cd` (**change directory**) consente di passare da una directory ad un'altra.

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

- `$ cd /prova` (**path assoluto**)

⇒ si passa alla directory prova;

- `$ cd prova` (**path relativo**)

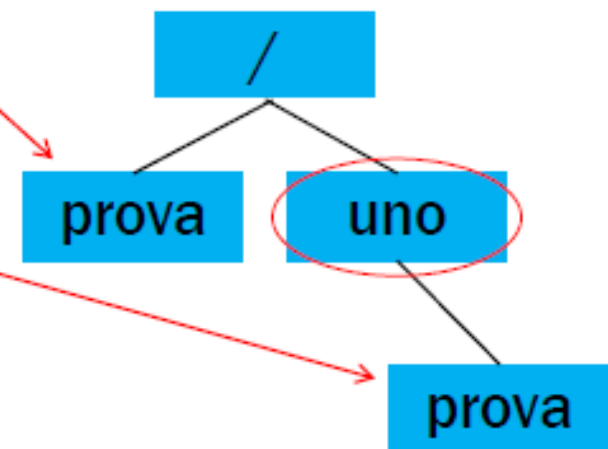
⇒ si passa alla directory prova;

- `$ 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
```

Comando ls (1/3)

`ls` (list segments) visualizza i nomi di file o il contenuto delle directory indicate.

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
```

Comando ls (2/3)

- 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 - .**

Comando ls (3/3)

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`

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

Comando man (1/2)

- Documentazione consultabile tramite il comando `man` (manual).
- Le pagine di tale documentazione sono raggruppate in sezioni omogenee in base al tipo di argomento trattato, ad esempio:
 - Sezione 1: comandi generali,
 - Sezione 2: chiamate di sistema,...

Comando man (2/2)

La sintassi generale è la seguente:

```
man nome_comando
```

per accedere alle pagine man di una sezione del manuale che non sia quella predefinita, è necessario specificare la sezione alla quale si è interessati.

Esempio:

- `$ man 1 printf`
- `$ man 3 printf`

Apropos e whatis

Altri due comandi di help disponibili in Unix sono:

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

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, ln

- `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 less e more

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

Comandi tail, head

- `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 (1/2)

La redirezione reindirizza 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 (2/2)

- **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.