

# Introduzione ai sistemi UNIX/Linux

M. Di Francesco, Vanessa Gardellin, Sara Volpi

Ilaria Giannetti, Daniele Migliorini

Dipartimento di Ingegneria dell'Informazione  
Università di Pisa

## Informazioni generali



- Pagina web del corso
  - <http://www.ing.unipi.it/~a008149/corsi/so/>
- Materiale:
  - Dispensa “**Amministrazione di un Sistema UNIX in Rete**” a cura di Giuseppe Anastasi, Andrea Passarella
  - Sezione “Materiale per le esercitazioni”

## 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

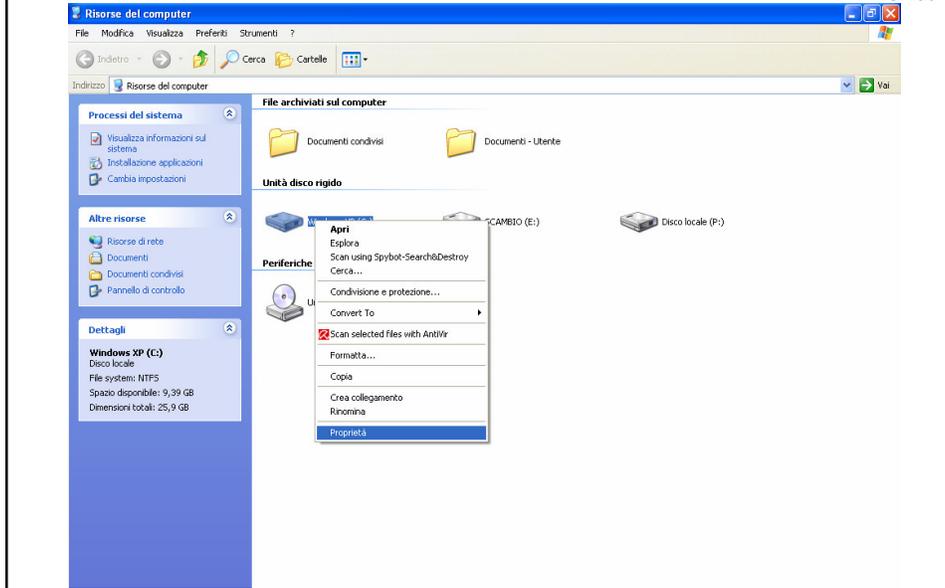
## Preparazione del disco



- 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

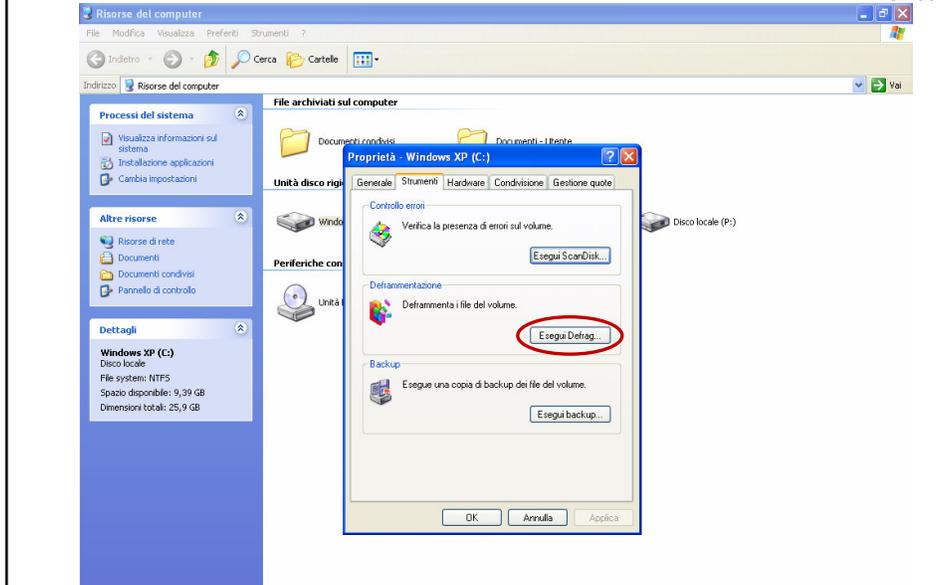
# Deframmentazione

1/3



# Deframmentazione

2/3





Volume	Stato sessione	File system	Capacità	Spazio disponibile	% spazio disponibile
Windows XP (C:)	Deframmentazione...	NTFS	26,00 GB	9,39 GB	36 %
SCAMBIO (E:)		FAT32	12,85 GB	4,68 GB	36 %

Stima utilizzo disco prima della deframmentazione:

Stima utilizzo disco dopo deframmentazione:

Analizza Deframmenta Sospendi Arresta Visualizza rapporto

File frammentati File contigui File fissi Spazio disponibile

Windows XP (C:) Deframmentazione in corso... 1% Compressione dei file in corso



- Partizioni primarie
  - specificate nella tabella delle partizioni del master boot record (MBR)
  - al più quattro partizioni
  - una sola avviabile
- 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

## 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

# Elementi fondamentali

## Introduzione al filesystem

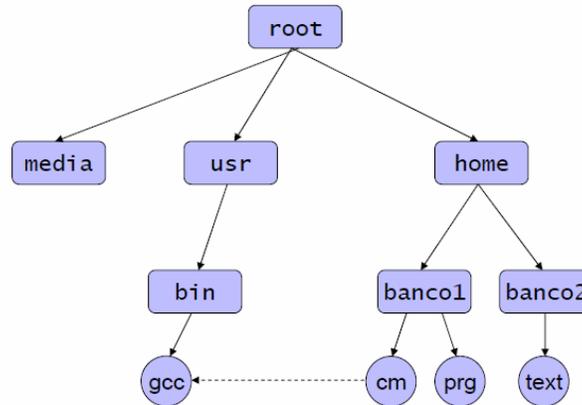


- Necessità di memorizzazione dei dati
- File
  - unità logica di memorizzazione dei dati
  - insieme di informazioni accessibili attraverso il nome del file stesso
- Cartella o directory
  - insieme di file e cartelle
- Filesystem
  - sistema di immagazzinamento e di organizzazione dei file

## 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

## Interprete dei comandi (shell) 1/2



- 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

```
ccna@iit-4:~/lab$
```

Diagram illustrating the components of the shell prompt `ccna@iit-4:~/lab$`:

- `ccna`: username
- `@iit-4`: host
- `:~/lab`: cartella corrente (current directory)
- `$`: cursore (cursor)



- 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



- 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

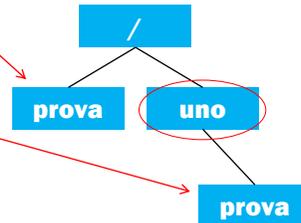


- `cd` (change directory) consente di passare da una directory ad un'altra.
- La sintassi generale di `cd` è la seguente:

```
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
```



- 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, 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.
- `ln [opzioni] sorgente destinazione`: crea collegamenti fisici o simbolici a file e directory.

## Comandi lettura da file



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



- `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 redirige 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`.



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.

Esercitazione



- Creare una cartella `temp` nella propria home
- Entrare nella cartella appena creata
- Creare due sottocartelle
  - `sorgente`
  - `sottolivello/destinazione`
- Creare nella cartella `sorgente` un file di nome `esempio` contenente la riga `contenuto`
- Aggiungere al file di nome `esempio` la riga `aggiunta`



- Creare nella cartella `sorgente` un file di nome `elenco` contenente l'elenco dei file e delle cartelle presenti nella root (`/`)
- Creare nella cartella `sottolivello/destinazione` il file `concat` prodotto dalla concatenazione dei file `esempio` ed `elenco`
- Visualizzare con il comando `less` le ultime tre righe del file `concat` (usare il comando `tail`)
- Cancellare la cartella `temp` e il suo contenuto

## Soluzione



```
mkdir temp
cd temp
mkdir sorgente
mkdir -p sottolivello/destinazione
echo "contenuto" > sorgente/esempio
echo "aggiunta" >> sorgente/esempio
ls / > sorgente/elenco
cat sorgente/esempio sorgente/elenco >
sottolivello/destinazione/concat
tail -n 3 sottolivello/destinazione/concat |
less
cd ..
rm -rf temp
```

## 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 open source:
    - ⇒ VirtualBox  
<http://www.virtualbox.org/>