

Organizzazione di Sistemi Operativi e Reti

Corso di Laurea Triennale in Ingegneria Informatica

Sara Lioba Volpi

Informazioni generali (1 di 3)



■ E-mail

- vanessa.gardellin@iet.unipi.it
- sara.volpi@iet.unipi.it

■ Pagina web

- <http://www2.ing.unipi.it/~o4003499>
- <http://www2.ing.unipi.it/~o4063499>

■ Ufficio

- Dipartimento di Ingegneria dell'Informazione,
Università di Pisa (polo A), via Diotisalvi 2, Pisa.

Informazioni generali (2 di 3)



- **Ricevimento**
 - su appuntamento (inviare un e-mail).

- **E-mail**
 - specificare **SEMPRE** come oggetto “OSOR”,
 - nell’e-mail devono essere specificati i problemi riscontrati.

Informazioni generali (3 di 3)



- **Pagina web del corso**
 - <http://www2.ing.unipi.it/~d8149/corsi/osor>
 - **N.B.** Nella pagina web del corso è presente un documento in cui sono sintetizzati gli argomenti più frequenti nelle domande degli studenti.
- **Esame**
 - consegna e discussione del progetto,
 - esame orale (parte pratica) relativa alle esercitazioni in laboratorio,
 - esame orale (parte teorica),
 - iscrizione obbligatoria su Hamasy.

Prima Esercitazione

- **Introduzione a Unix,**
- **Shell,**
- **Filesystem,**
- **Comandi di base,**
- **Altri comandi.**

Introduzione ai sistemi Unix

Breve storia di Unix



- **1969** - Prima edizione di UNIX sviluppata da Ken Thompson presso i Bell Laboratories.
- **1973** - Thompson e Dennis 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 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.



- **FreeBSD** è un sistema operativo libero Unix-like, rilasciato per la prima volta nel **1993**.
- La maggior parte di FreeBSD è rilasciato sotto copyright BSD, il resto è sotto la GNU GPL o altre licenze open-source.
- **Varianti BSD:**
 - NetBSD,
 - OpenBSD.



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.

- **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 FreeBSD

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 con es.
 - ⇒ Partition Magic,
 - ⇒ gparted (<http://gparted.sourceforge.net>),
 - consigliata deframmentazione preventiva.

Partizionamento (1 di 2)



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 (2 di 2)

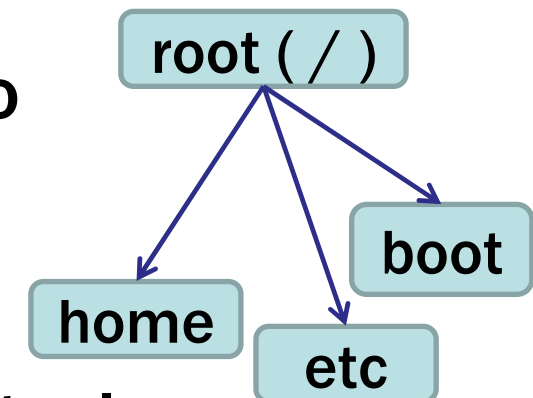


Partizioni FreeBSD:

- le *partizioni primarie* sono chiamate *slice*,
- le *partizioni logiche* sono chiamate *partizioni*,
- il S.O. deve essere installato in una **partizione primaria (non estesa)**.

- 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.



Elementi fondamentali: file system

Introduzione ai filesystem



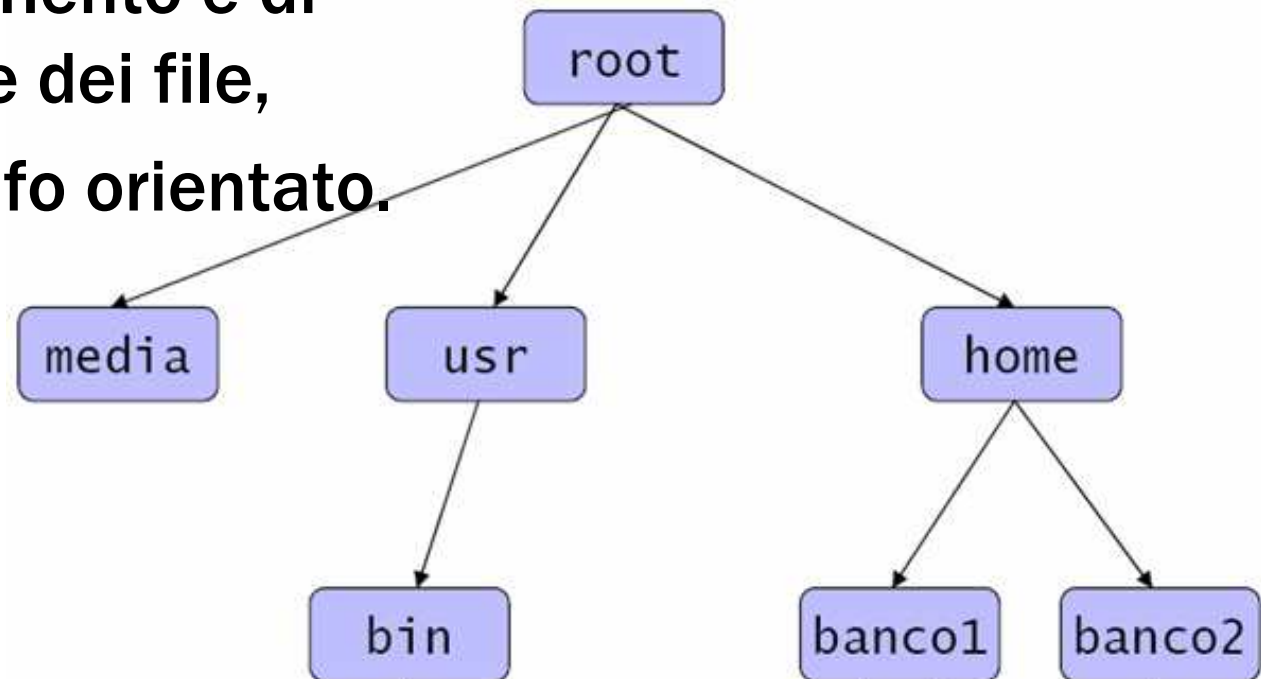
- **Necessità di memorizzazione dei dati.**
- **File**
 - unità logica di memorizzazione dei dati,
 - insieme di informazioni accessibili attraverso il nome del file stesso.
- **Directory**
 - insieme di file.

Organizzazione del filesystem (1 di 3)



■ Filesystem

- sistema di immagazzinamento e di organizzazione dei file,
- struttura a grafo orientato.



Organizzazione del filesystem (2 di 3)



Sottocartelle della cartella principale / (root):

- **/bin** programmi eseguibili dall'utente,
- **/dev** file dei device,
- **/etc** file di configurazione del sistema,
- **/lib** librerie condivise,
- **/opt** software opzionale,
- **/tmp** file temporanei,

Organizzazione del filesystem (3 di 3)



- **/usr** **comandi e strumenti dell'utente,**
- **/var** **file di dimensione variabile,**
- **/root** **home dell'utente root,**
- **/home** **cartelle personali degli utenti,**
- **/home/ut** **home di un generico utente “ut”.**

Elementi fondamentali: shell



Accesso al sistema

Login iniziale:

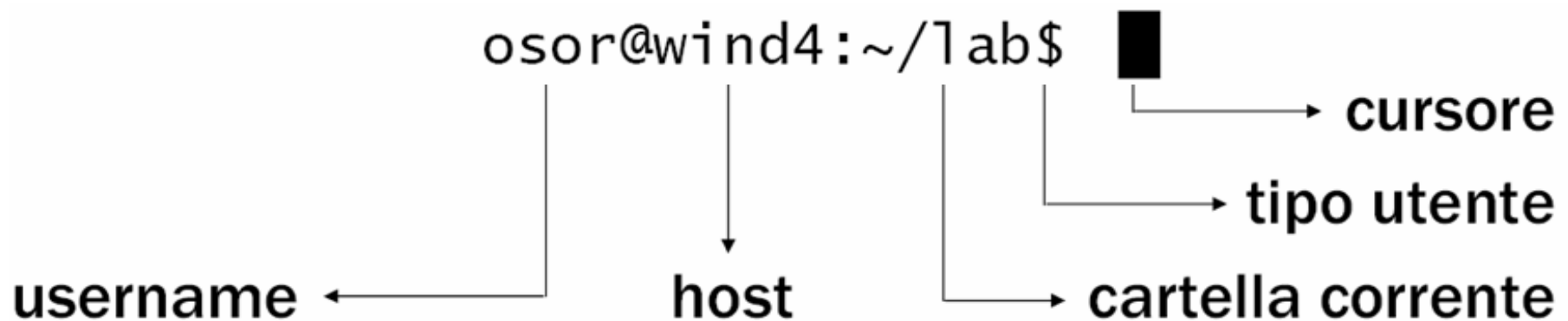
- `login: nomeUtente`
- `Password: password`

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

Shell (1 di 2)



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



Shell (2 di 2)



- **Funzioni**
 - **autocompletamento (tasto TAB),**
 - **history (freccia SU/GIU).**
- **I sistemi UNIX-like sono case sensitive,**
 - **esempio**
 - ⇒ file1, File1, FILE1, FiLe1, sono file diversi.

Comandi di base

Comandi di base



- `Ctrl + d` : **logout**,
- `login`
- `Alt + F1 (F2, F3, ...)` : **passaggio ad altri terminali.**

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).**

Comando cd (1 di 2)



- **path assoluto:** si parte dalla directory root /.
- **path relativo:** si parte dalla directory corrente.

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

Comando cd (2 di 2)



Esempio

- Sia uno la directory corrente, se si digita:

- `$ cd /prova` (path assoluto)

⇒ si passa alla directory prova;

- `$ cd prova` (path relativo)

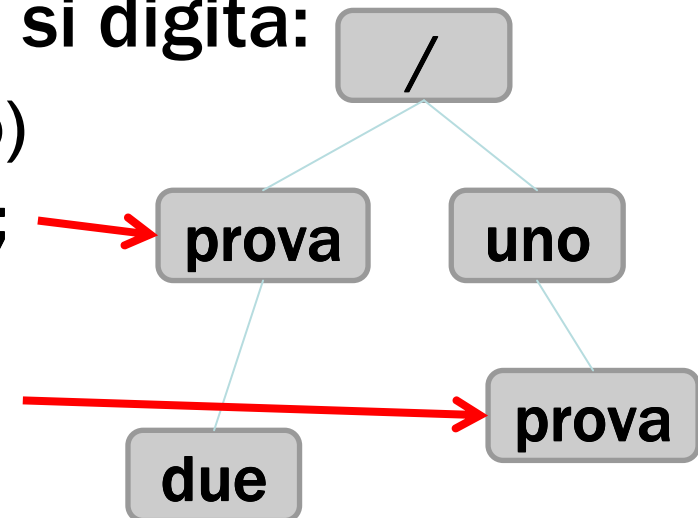
⇒ si passa alla directory prova;

- `$ cd ~` o `cd`

⇒ Si passa alla directory home dell'utente;

- `$ cd ..`

⇒ Si passa alla directory genitrice 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 di 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 di 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 di 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`

Comandi di help

Comando man (1 di 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 di 2)



La sintassi generale è la seguente:

```
man nome_comando
```

ma, 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`

Comandi apropos, 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 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.**

Lettura file

Comandi less, more



- `less [opzioni] [file]..`
- `more [opzioni] [file]..`

Utilizzati per leggere file di puro testo.

Operazioni sui file

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 e pipeline

Redirezione (1 di 2)



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 (2 di 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`.

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.**

mtools

- **mtools** è un insieme di comandi per la gestione dei dati su floppy disk.

<code>mcopy /bin/prova a:</code>	Copia il file prova da /bin al dischetto (a:).
<code>mdir a:</code>	Visualizza i file del dischetto (a:).
<code>mcopy a:\prova /home</code>	Copia il file prova dal dischetto (a:) in /home.
<code>mdel a:\prova</code>	Elimina il file prova dal dischetto (a:).

Esercizio

Esercizio (1 di 2)



1. Creare una cartella temp nella propria home.
2. Entrare nella cartella appena creata.
3. Creare due sottocartelle:
 1. Sorgente,
 2. sottolivello/destinazione.
4. Creare nella cartella sorgente un file di nome esempio contenente la riga “contenuto” (usare il comando echo).
5. Aggiungere al file di nome esempio la riga “aggiunta” (usare il comando echo).

Esercizio (2 di 2)



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

Soluzione



1. `cd ~`
`mkdir temp`
2. `cd temp`
- 3.1 `mkdir sorgente`
- 3.2 `mkdir -p sottolivello/destinazione`
4. `echo "contenuto" > sorgente/esempio`
5. `echo "aggiunta" >> sorgente/esempio`
6. `ls / > sorgente/elenco`
7. `cat sorgente/esempio sorgente/elenco >`
`sottolivello/destinazione/concat`
8. `tail -n 3 sottolivello/destinazione/concat | less`
9. `cd ..`
`rm -rf temp`