

Sistemi Operativi ed Estensioni dell'architettura

Capitolo 3 del testo

Alberto Policriti



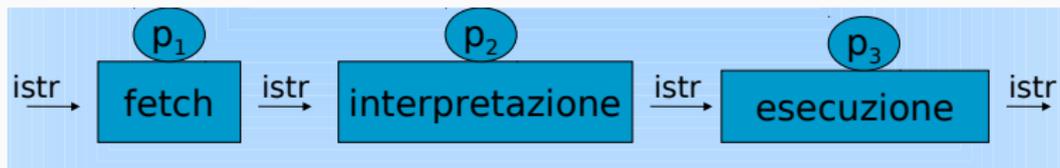
31 Ottobre, 2019

Limiti

1. Non implementa alcun tipo di parallelismo
2. La CPU trasferisce i dati sui bus in modo sequenziale

... sono presenti limiti facilmente superabili senza stravolgere il sistema.

- Processori dedicati e co-processori (matematici)
- **Pipelining**: le varie fasi vengono affidate a diverse componenti che operano contemporaneamente.



- Gerarchie di memorie e **caches**
- Processori dedicati all'I/O
- Architetture **multi-processore**
- ...

Interfacce Utente

Software Applicativo

Software di base

FirmWare (BIOS)

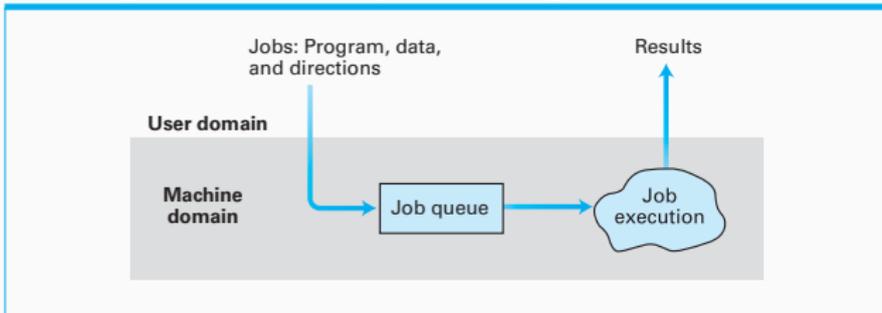
Unità logiche (CPU, Memoria, Periferiche, etc)

Circuiti Elettronici Digitali Hardware

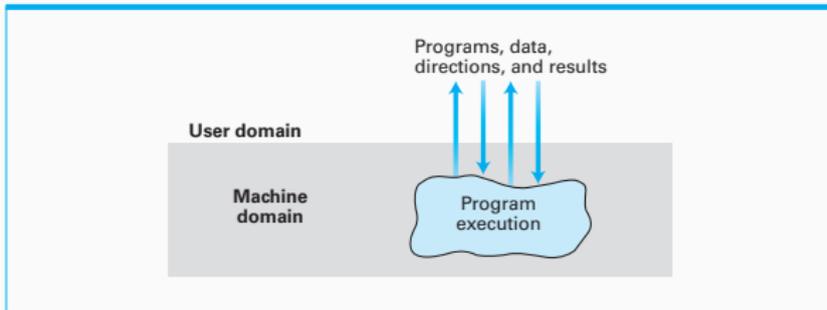
Nomi

- Windows
- Unix
- MacOS
- Linux
- ...

Una volta ... jobs e batch processing



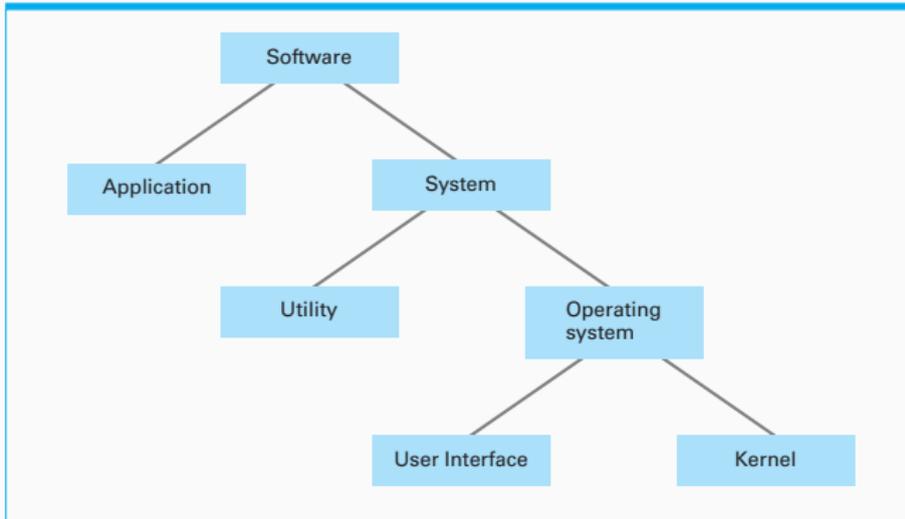
... poi ... interactive processing



... poi

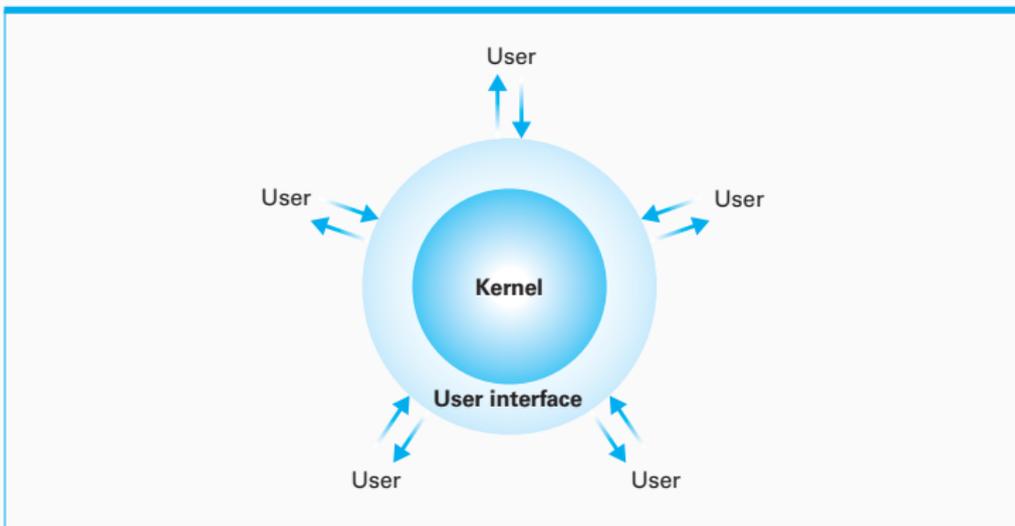
- interactive processing
- real-time processing
- time-sharing
- multiprogramming/multitasking
- load-balancing
- scaling
- ...
- embedded systems

Terminologia (non rigida)



Componenti

- interfacce utente: shells e graphical user interface (GUI)
- window manager



Il file manager

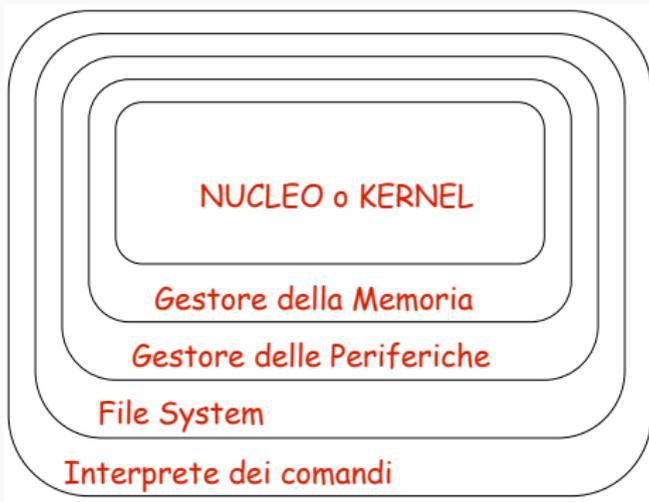
Lo meglio vedremo più avanti

- Software che opera direttamente sull'hardware
- Definisce una gerarchia di **macchine virtuali**
- E' composto da vari *strati* che sono definiti in termini di funzioni fornite all'utente (“buccia di cipolla”)

compiti

- Semplificare agli utenti l'accesso e l'utilizzo delle risorse
- Presentare ai processi le risorse in modo astratto
- Arbitrare
- Proteggere

strati



Nucleo

E' lo strato del S.O. che gestisce i **processi**

- Un processo è la coppia costituita da un programma e dall'insieme dei valori contenuti in memoria (RAM e registri) per la sua esecuzione
- i processi sono in **esecuzione** / **attesa** / **ready**

Gestore della memoria

E' lo strato del S.O. che

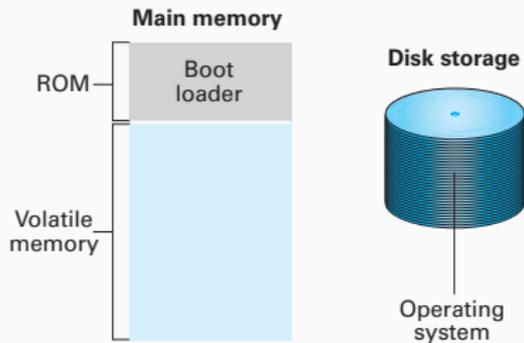
- ripartisce la RAM fra i programmi in esecuzione
- alloca la memoria (ripartizione, assegnazione e conversione degli indirizzi di memoria)
- gestisce la *segmentazione* (blocchi di lunghezza variabile) e la *paginazione* (blocchi di lunghezza fissa)

Gestore delle periferiche

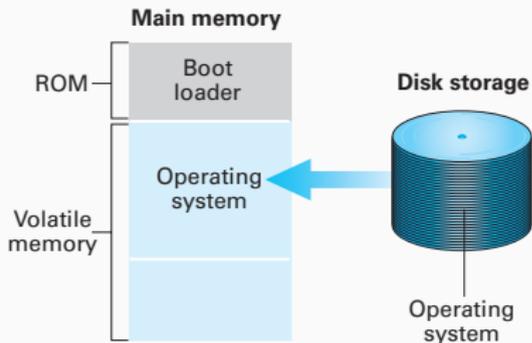
E' lo strato del S.O. che

- si occupa dell'interfaccia con le periferiche
- "nasconde" all'utente le caratteristiche dell'hw per le operazioni di I/O
- gestisce le code per l'accesso alle periferiche

Come inizia il lavoro? Booting



Step 1: Machine starts by executing the boot loader program already in memory. Operating system is stored in mass storage.



Step 2: Boot loader program directs the transfer of the operating system into main memory and then transfers control to it.

gli strati più importanti (per noi)

- File System (permette di utilizzare la memoria di massa)
- Interprete dei comandi (permette di sfruttare a pieno le potenzialità della macchina)

File System e sua gestione

E' lo strato del S.O. che si occupa di leggere, scrivere, ... *gestire* i file nella memoria di massa

- creazione
- rinomina, movimento, privilegi, ...
- resa trasparente dell'hw per l'utente

Memoria virtuale.

Example

L'interfaccia ed i comandi per accedere al file system di questo computer

- i files devono essere manipolabili in modo **trasparente** all'utente rispetto all'hw
- **directories (cartelle)**: servono a dare una struttura ad *albero* al file system (sono i nodi *interni* all'albero)

- il file system contiene directories in cui è memorizzato il sistema operativo e tutti i moduli ad esso necessari

Example

La directory System in questo pc

Unix

tutto è un file

home

- In un sistema multi-utente è la directory che definisce l'area di memoria utilizzata da un singolo utente
- è possibile creare delle sotto-directory
- vi sono diversi programmi per “navigare” nel file system

Example

I programmi Finder, X, Gnome, KDE, ...

- Le finestre sono i nodi dell'albero
- “entrare” in una finestra equivale a scendere nell'albero
- i files vengono rappresentati mediante icone e:
 - hanno un suffisso
 - il suffisso ne identifica il tipo
 - l'icona ne rappresenta graficamente il tipo
- la nozione di alias diventa particolarmente importante ed utile

- I programmi sono contenuti in files
- spesso il tipo dei file che sono programmi è molto facilmente identificabile
- “aprire” un file contenente un programma equivale a lanciare il programma stesso

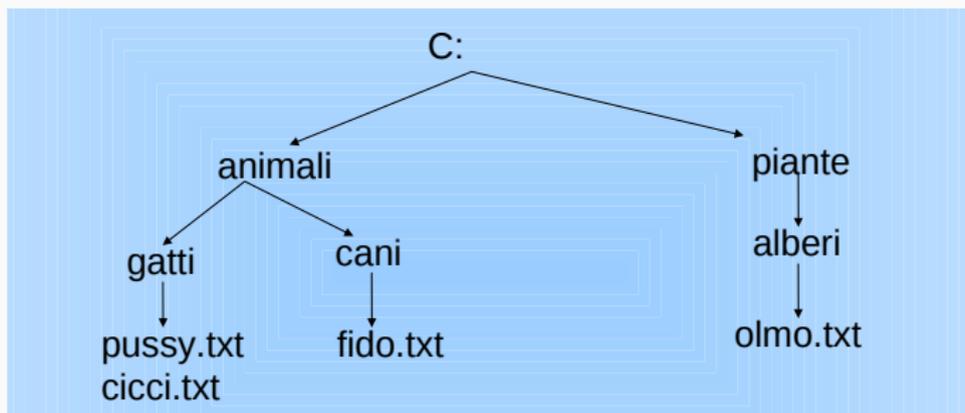
Domanda

le performances di un computer variano a seconda di quanti programmi ho installato?

- Ad ogni file possono essere associate particolari proprietà
- i file sono identificati dai loro pathname (nomi completi)
- ai file possono essere associati dei **link**

Example

1. A cosa corrisponde Desktop su questo pc?
2. A cosa corrisponde Documenti?



pathname (logico) di `cicci.txt`

```
C:\animali\gatti\cicci.txt
```

- verticale (all'indietro): su diverse versioni
- orizzontale: relativamente a diversi o.s., applicativi, ...

Importantissima e spesso garantita solo “sulla carta”

- Ogni sistema operativo mette a disposizione una *shell* di comandi
- I comandi servono anche per manipolare il file system
- I comandi possono essere dati su linea (interfaccia testuale, **prompt**) o eseguibili tramite mouse (interfaccia grafica, **finestre**)
- I comandi vengono eseguiti in un **contesto**

Example

Qualche comando in contesti diversi su questo pc

- E' importante
- molto spesso è ignorato
- altrettanto spesso può risultare troppo vago o troppo specifico

... bisogna imparare ad usarlo (e ad usare la rete)

Example

L' "Help" dall'interfaccia grafica e da linea di comando di questo pc.

I **percorsi relativi** iniziano con: il nome di una directory; con i simboli “./”.

Esempi:

Scrivania/Sport

./Documenti

I **percorsi assoluti** iniziano con: il simbolo “/”; i simboli “//”; i simboli “~/”.

Esempi:

/home/mrossi

//usr

~/Documenti

Alcuni comandi della shell di Linux

pwd (*print working directory* = “*stampa la directory corrente*”) visualizza il percorso assoluto della directory corrente (directory di lavoro).

uso

pwd

esempio

→ **[mrossi@localhost Documenti]\$ pwd ↵ (invio)**

→ **/home/mrossi/Documenti**

→ **[mrossi@localhost Documenti]\$**

→ **[mrossi@localhost ~]\$ pwd ↵ (invio)**

→ **/home/mrossi/**

→ **[mrossi@localhost ~]\$**

← **NOTA:** nella visualizzazione completa del percorso non appare la tilde, “~”, come ultimo simbolo perché viene automaticamente sostituita dal nome della “home directory” dell’attuale utente, in questo caso “mrossi”.

Alcuni comandi della shell di Linux

ls (*list segments* = “*elenca segmenti*”)

visualizza il contenuto della directory indicata dal percorso. Se non viene indicato nessun percorso verrà visualizzato il contenuto della directory corrente (directory di lavoro).

uso

ls [*opzioni*] [*percorso*]

opzioni

-l visualizza una lunga e dettagliata lista di informazioni per ogni file o directory.

-la visualizza anche i file o le directory nascoste.

(le varie opzioni possono essere combinate insieme, senza un preciso ordine, come ad esempio: **-al** oppure **-la**)

percorso

Inserendo un percorso permette di vedere il contenuto di una directory diversa da quella corrente, senza spostarsi.

esempio

→ **[mrossi@localhost ~]\$ ls** ↵ (invio)

→ Documenti Immagini Musica prova.txt Scaricati

→ **[mrossi@localhost ~]\$**

→ **[mrossi@localhost ~]\$ ls -l** ↵ (invio)

→ drwxrwxr-x. 1 mrossi mrossi 4096 14 sett 19:50 Documenti

→ drwxrwxr-x. 1 mrossi mrossi 4096 29 nov 00:50 Immagini

→ drwxrwxr-x. 2 mrossi mrossi 4096 14 sett 19:50 Musica

→ -rwxrwxr-x. 1 mrossi mrossi 4096 15 dic 01:32 prova.txt

→ drwxrwxr-x. 2 mrossi mrossi 4096 14 sett 19:50 Scaricati

→ **[mrossi@localhost ~]\$**

→ **[mrossi@localhost ~]\$ ls Scrivania** ↵ (invio)

→ Cartella prova1.txt prova2.txt

→ **[mrossi@localhost ~]\$** ← *NOTA: dopo l'esecuzione del comando, la directory di lavoro è rimasta invariata, lo si può vedere dalle informazioni visualizzate prima del prompt. Nell'esempio non ci si è spostati dalla "home directory", indicata con il simbolo tilde (~).*

Alcuni comandi della shell di Linux

INFORMAZIONI VISUALIZZATE CON IL COMANDO `ls -l`

```
drwxrwxr-x. 2 mrossi mrossi 4096 14 sett 19:50 Scaricati
↑           ↑ ↑         ↑         ↑         ↑         ↑
Col 1      Col 2 Col 3   Col 4     Col 5     Col 6         Col 7
```

Col 1 - permessi

```
d rwx rwx r-x
 1  2  3  4
```

1. Indica il tipo di oggetto.

d = directory

- = file

l = link

b = block o char device ...ecc

2. permessi per l'utente

3. permessi per il gruppo dell'utente

4. permessi per tutti gli utenti

} r = lettura / w = scrittura / x = esecuzione / - = permesso negato

Col 2 - numero di link all'oggetto

Col 3 - la directory o il proprietario del file

Col 4 - gruppo appartenuto dall'utente

Col 5 - la dimensione di ogni file in byte

Col 6 - data e ora di ultima modifica del file

Col 7 - il nome del file o della directory

Alcuni comandi della shell di Linux

cd (*change directory* = “cambia directory”)
cambia la directory corrente (directory di lavoro).

USO

cd [percorso]

esempio (per tutti gli esempi riportati fai riferimento all'immagine del filesystem in prima pagina)

→ [mrossi@localhost Documenti]\$ cd ↵ (invio) ← NOTA: se non viene indicato nessun percorso, normalmente va nella “home directory” dell'utente. Equivale al comando “cd ~”.

→ [mrossi@localhost Documenti]\$ cd ~ ↵ (invio) ← NOTA: il simbolo tilde (~), dopo il comando, indica che la nuova directory di lavoro è la “home directory” dell'utente.

→ [mrossi@localhost Documenti]\$ cd / ↵ (invio) ← NOTA: il simbolo slash (/), dopo il comando, indica che la nuova directory di lavoro è la directory di root o radice.

→ [mrossi@localhost Documenti]\$ cd .. ↵ (invio) ← NOTA: il simboli punto punto (..), dopo il comando, indica che la nuova directory di lavoro è la directory madre della directory corrente. In questo esempio la directory madre della directory “Documenti” è la “home directory” dell'utente.

→ [mrossi@localhost Documenti]\$ cd /usr ↵ (invio)

→ [mrossi@localhost usr]\$

→ [mrossi@localhost usr]\$ cd ~/Scrivania ↵ (invio)

→ [mrossi@localhost Scrivania]\$

Alcuni comandi della shell di Linux

mkdir (*make directory* = “crea directory”)

crea una nuova directory all'interno della directory indicata dal percorso. Se non viene indicato nessun percorso verrà creata nella directory corrente (directory di lavoro).

uso

mkdir [percorso][nome nuova directory]

esempio

→ [mrossi@localhost ~]\$ **mkdir Lavoro** ↵ (invio) ← NOTA: crea una directory di nome “Lavoro” all'interno della directory corrente, la “home directory” dell'utente, indicata con la tilde (~).

→ [mrossi@localhost ~]\$ **mkdir Scrivania/Sport** ↵ (invio) ← NOTA: crea una directory di nome “Sport” all'interno della directory indicata dal percorso, in questo caso “Scrivania”.

→ [mrossi@localhost ~]\$ **mkdir ~/Scrivania/Sport/Calcio** ↵ (invio) ← NOTA: crea una directory di nome “Calcio” all'interno della directory indicata dal percorso, in questo caso “Sport”.

Alcuni comandi della shell di Linux

rmdir (*remove directory* = “rimuovi directory”)

rimuove la directory indicata dal percorso. Se non viene indicato nessun percorso verrà cercata e consecutivamente cancellata nella directory corrente (directory di lavoro).

uso

rmdir [percorso][nome directory da rimuovere]

esempio

→ [mrossi@localhost ~]\$ **rmdir Lavoro** ↵ (invio) ← *NOTA: rimuove la directory di nome “Lavoro” presente all’interno della directory corrente, la “home directory” dell’utente, indicata con la tilde (~).*

→ [mrossi@localhost /]\$ **rmdir ~/Scrivania/Sport/Calcio** ↵ (invio) ← *NOTA: rimuove la directory di nome “Calcio” che si trova nella directory indicata dal percorso, in questo caso “Sport”.*

touch ()

crea un file vuoto. Se non viene indicato nessun percorso il file verrà creato nella directory corrente (directory di lavoro).

USO

touch [*percorso*][*nome file*]

Alcuni comandi della shell di Linux

esempio

→ [mrossi@localhost ~]\$ touch ordini.txt ↵ (invio)

→ [mrossi@localhost ~]\$

← NOTA: crea un file di nome "ordini.txt" all'interno della directory corrente, la "home directory" dell'utente, indicata con la tilde (~).

→ [mrossi@localhost ~]\$ touch **Scrivania**/elenco.txt ↵ (invio)

→ [mrossi@localhost ~]\$

← NOTA: crea un file di nome "elenco.txt" all'interno della directory indicata dal percorso, in questo caso "Scrivania".

→ [mrossi@localhost /]\$ touch **~/Documenti**/spese.txt ↵ (invio)

→ [mrossi@localhost /]\$

← NOTA: crea un file di nome "spese.txt" all'interno della directory indicata dal percorso, in questo caso "Documenti".

Alcuni comandi della shell di Linux

rm (*remove* = “rimuovi”)

rimuove file o directory (con l'opzione *-r*) indicata dal percorso. Se non viene indicato nessun percorso verrà cercata e consecutivamente cancellata nella directory corrente (directory di lavoro).

uso

rm [opzioni] [percorsol][nome file o directory da rimuovere]

opzioni

-r elimina in modo ricorsivo tutti i file e/o le directory presenti all'interno della directory da cancellare.

esempio

→ [mrossi@localhost ~]\$ **rm ordini.txt** ↵ (invio) ← NOTA: rimuove il file di nome “ordini.txt” presente all'interno della directory corrente, la “home directory” dell'utente, indicata con la tilde (~).

→ [mrossi@localhost /]\$ **rm ~/Documenti/spese.txt** ↵ (invio) ← NOTA: rimuove il file di nome “spese.txt” che si trova nella directory indicata dal percorso, in questo caso “Documenti”.

→ [mrossi@localhost /]\$ **rm -r ~/Scrivania/Sport** ↵ (invio) ← NOTA: rimuove la directory “Sport” e con essa la directory “Calcio” presente al suo interno, grazie all'opzione “-r”, che si trova nella directory indicata dal percorso, in questo caso “Scrivania”.

Alcuni comandi della shell di Linux

mv (*move* = “sposta”)

sposta o rinomina un file o una directory.

uso

mv [*percorso1*][nome file o directory da spostare o da rinominare] [*percorso di destinazione*][*nuovo nome*]

esempio

→ [mrossi@localhost ~]\$ mv ordini.txt **Scrivania** ↵ (invio) ← NOTA: sposta il file di nome “ordini.txt”, presente nella directory corrente, nella directory “Scrivania”.

→ [mrossi@localhost Documenti]\$ mv ~/Scrivania/ordini.txt ↵ (invio) ← NOTA: sposta il file di nome “ordini.txt”, presente nella directory “Scrivania”, nella directory “Documenti”.

Alcuni comandi della shell di Linux

- `[mrossi@localhost Scrivania]$ mv spese.txt spese_ferie.txt ↵ (invio)` ← NOTA: rinomina il file "spese.txt" con il nuovo nome: "spese_ferie.txt".
- `[mrossi@localhost Scrivania]$`
- `[mrossi@localhost /]$ mv ~/Scrivania/ordini.txt ~/Documenti/lista.txt ↵ (invio)` ← NOTA: sposta il file "ordini.txt" dalla directory "Scrivania" alla directory "Documenti" e contemporaneamente lo rinomina in "lista.txt".
- `[mrossi@localhost /]$`
- `[mrossi@localhost /]$ mv ~/Scrivania/Sport/Calcio ~/Scrivania/Sport/Calcetto ↵ (invio)` ← NOTA: rinomina la directory "Calcio" con il nuovo nome "Calcetto".
- `[mrossi@localhost /]$`
- `[mrossi@localhost Sport]$ mv Calcetto ~/Documenti ↵ (invio)` ← NOTA: sposta la directory "Calcetto" nella directory "Documenti".
- `[mrossi@localhost Sport]$`

Alcuni comandi della shell di Linux

cp (*copy* = "copia")

effettua la copia di un file o una directory. E' possibile assegnare alla copia un nuovo nome. Se non vengono specificati dei percorsi, sia quello dell'oggetto da copiare che quello di destinazione, o se i percorsi sono uguali, la copia verrà creata nella stessa directory e, in questo caso, è necessario assegnare alla copia un nuovo nome.

uso

cp [*percorso1*][nome file o directory da spostare o da rinominare] [*percorso di destinazione*][nuovo nome]

esempio

→ [mrossi@localhost ~]\$ cp ordini.txt Scrivania ↵ (invio) ← NOTA: fa una copia del file di nome "ordini.txt", presente nella directory corrente, nella directory "Scrivania".
→ [mrossi@localhost ~]\$

→ [mrossi@localhost Documenti]\$ cp ~/Scrivania/ordini.txt ↵ (invio) ← NOTA: fa una copia del file di nome "ordini.txt", presente nella directory "Scrivania", nella directory "Documenti".
→ [mrossi@localhost Documenti]\$

→ [mrossi@localhost Scrivania]\$ cp spese.txt spese_ferie.txt ↵ (invio) ← NOTA: fa una copia del file "spese.txt" nella medesima directory di lavoro assegnando ad essa un nuovo nome: "spese_ferie.txt".
→ [mrossi@localhost Scrivania]\$

→ [mrossi@localhost /]\$ cp ~/Scrivania/ordini.txt ~/Documenti/lista.txt ↵ (invio) ← NOTA: fa una copia del file "ordini.txt" nella directory "Scrivania", nella directory "Documenti" assegnandogli un nuovo nome: "lista.txt".
→ [mrossi@localhost /]\$

→ [mrossi@localhost /]\$ cp ~/Scrivania/Sport/Calcio ~/Scrivania/Sport/Calcetto ↵ (invio) ← NOTA: fa una copia della directory "Calcio" nella medesima directory di lavoro assegnandogli un nuovo nome: "Calcetto".
→ [mrossi@localhost /]\$

→ [mrossi@localhost Sport]\$ cp Calcetto ~/Documenti ↵ (invio) ← NOTA: fa una copia della directory "Calcetto" nella directory "Documenti".
→ [mrossi@localhost Sport]\$