

## Corso di Sistemi Operativi a.a. 2002

Tutor: **Paolo Torroni**

e-mail: [ptorroni@deis.unibo.it](mailto:ptorroni@deis.unibo.it)

tel. 051-20-93086

Ufficio: Laboratorio di Informatica Avanzata (LIA) nel Lab1 (edificio DEIS)

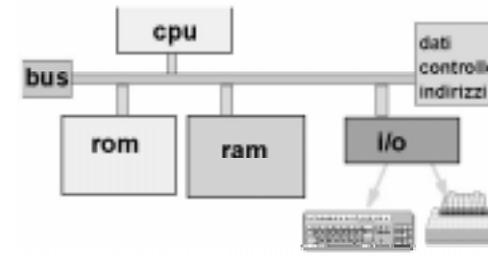
Siti per il materiale e le informazioni sul corso:

<http://www.citam.unibo.it/Tecnopolo/InfeAuto.html#Esercitazioni>

<http://www-lia.deis.unibo.it/Courses/Teledid/>

## ELABORATORE ELETTRONICO

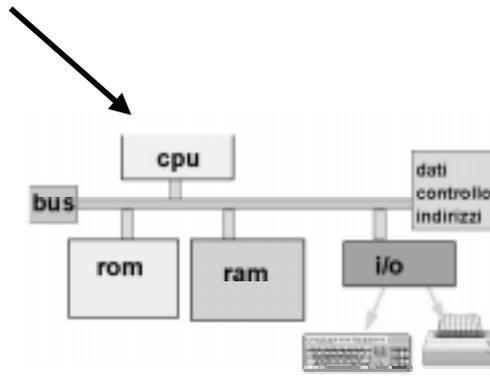
La parte Hardware e' composta da un insieme di *unità funzionali*



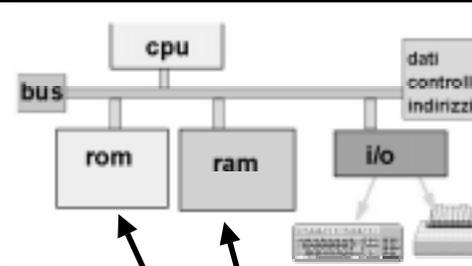
## CPU

CPU (Central Processing Unit), o Processore

**CPU:** Svolge le elaborazioni e il trasferimento dei dati, cioè *esegue i programmi*



## MEMORIA CENTRALE



### RAM & ROM

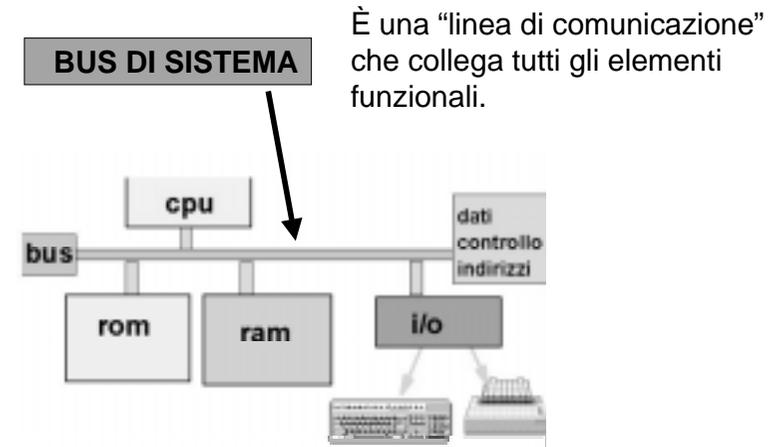
- Dimensioni relativamente limitate
- Accesso molto rapido

**RAM (Random Access Memory), e ROM (Read Only Memory)**  
Insieme formano la **Memoria centrale**

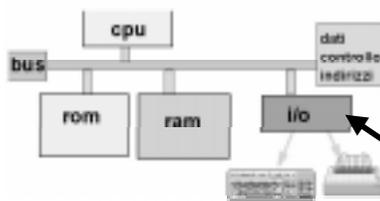
## MEMORIA CENTRALE (II)

- **RAM è volatile** (perde il suo contenuto quando si spegne il computer)
  - usata per memorizzare dati e programmi
- **ROM è persistente** (mantiene il suo contenuto quando si spegne il computer) ma il suo **contenuto è fisso e immutabile**
  - usata per memorizzare programmi di sistema

## BUS DI SISTEMA



## DISPOSITIVI I/O



Sono usate per far comunicare il computer con l'esterno (in particolare con l'utente)

### UNITÀ DI INGRESSO / USCITA (I/O)

- Tastiera e Mouse
- Video e Stampante
- Scanner
- Tavoleta grafica
- **Dispositivi di memoria di massa** (Dischi, CD, Nastri)
- ...

## TECNOLOGIA DIGITALE

CPU, memoria centrale e dispositivi sono realizzati con **tecnologia elettronica digitale**.

Dati ed operazioni vengono codificati a partire da due valori distinti di grandezze elettriche:

- tensione alta ( $V_H$ , 5V)
- tensione bassa ( $V_L$ , 0V)

A tali valori vengono convenzionalmente **associate le due cifre binarie 0 e 1**

## TECNOLOGIA DIGITALE (segue)

Dati ed operazioni vengono codificati tramite **sequenze di bit**

**01000110101 ....**

CPU è in grado di operare soltanto in aritmetica binaria, effettuando operazioni *elementari* :

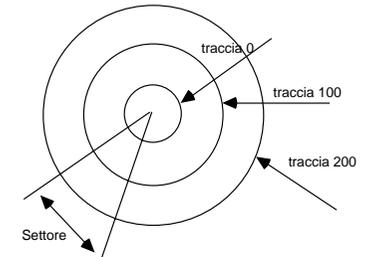
- somma e differenza
- scorrimento (shift)
- ...

Lavorando direttamente sull'hardware, l'utente è **forzato a esprimere i propri comandi a livello della macchina, tramite sequenze di bit.**

## Esempio: DISCHI MAGNETICI

Un disco consiste in un certo numero di **piatti** con **due superfici** che ruotano attorno ad un perno centrale.

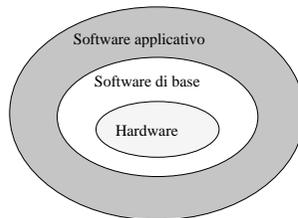
Ogni superficie dispone di una propria **testina di lettura / scrittura**.



Le superfici sono organizzate in **cerchi concentrici (tracce)** e in **spicchi** di ugual grandezza (**settori**). Ogni blocco del disco necessario per il trasferimento dei dati è identificato dalla terna {superficie, traccia, settore}

## SOFTWARE

**Software:** insieme complesso di programmi che vengono eseguiti dal sistema.

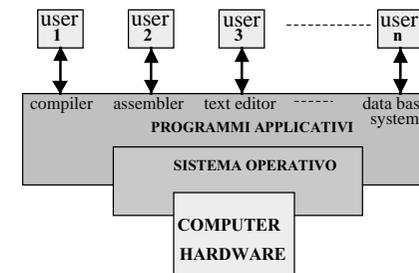


**Distinzione fra:**

- Software di base (es. Sistema Operativo)
- Software applicativo

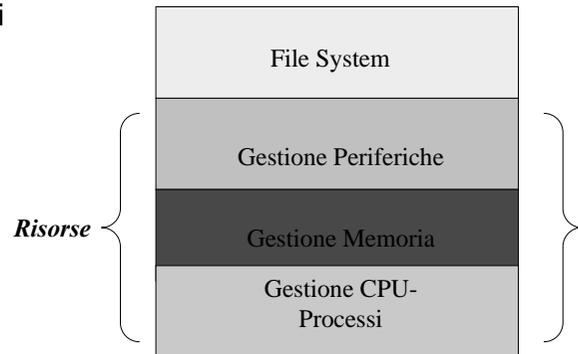
## IL SISTEMA OPERATIVO

Insieme di programmi che opera *al di sopra della macchina fisica, mascherandone le caratteristiche e fornendo agli utenti funzionalità di alto livello.*



## FUNZIONI DEL SISTEMA OPERATIVO

- **Gestione delle Risorse** per rendere efficiente l'uso delle risorse sia hardware che software
- Fornisce all'utente l'astrazione di **Macchina Virtuale** per semplificare l'uso del sistema di calcolo da parte degli utenti



## FUNZIONI DEL SISTEMA OPERATIVO

Qualsiasi operazione di accesso a risorse della macchina implicitamente richiesta dall'utente **viene eseguita dal S.O.**

Attraverso il S.O. il livello di interazione fra utente ed elaboratore viene elevato:

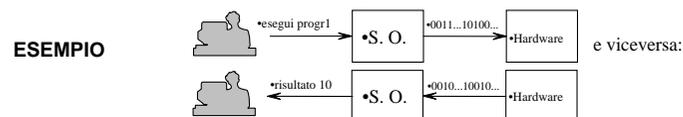
- senza S.O.: sequenze di bit
- con S.O.: comandi, programmi, dati

Il S.O. **traduce le richieste dell'utente** in opportune **sequenze di istruzioni**, a loro volta trasformate in **valori e impulsi elettrici** per la macchina fisica.

## ESEMPIO

Il S.O. permette per esempio

- accesso a memoria centrale
- accesso ai dischi
- I/O verso video, tastiera, ...



<u>Utente:</u> "esegui progr1"	<u>Sistema Operativo:</u> - input da tastiera - ricerca codice di "progr1" su disco - carica in memoria centrale codice e dati <elaborazione>
<u>Utente:</u> "stampa 10"	<u>Sistema Operativo:</u> - output su video

## GESTIONE DELLE RISORSE

- **Assegnazione delle risorse** del sistema di calcolo ai programmi di utente in base al tipo di richiesta e agli obiettivi da raggiungere (**uso efficiente delle risorse**)

- **Risorse Hw e Sw:** tempo di CPU, memoria, I/O, etc..

- \* risoluzione di **conflitti** nell'uso delle risorse
- \* scelta dei **criteri** con cui assegnare una risorsa

- Disponibilità di **apposite operazioni (system calls)** per la gestione delle risorse

## MACCHINA VIRTUALE

Astrazione per semplificare l'uso delle risorse e nascondere i dettagli implementativi.

Esempio: **Controllore di un floppy disk**

- numerosi comandi (*lettura, scrittura, movimento del braccio, formattamento delle tracce, etc..*)
- ogni comando ha più parametri (*indirizzo del blocco, numero di settori per traccia, etc..*)
- numerose condizioni di stato e di errore al completamento del comando



necessità di nascondere all'utente i dettagli hardware legati al particolare dispositivo. Uso di operazioni del **S.O. (file system)**

## CLASSIFICAZIONE dei S.O.

In base al numero di utenti:

- **Mono-utente (*mono-user*)**: un solo utente alla volta può utilizzare il sistema
- **Multi-utente (*multi-user*)**: più utenti possono interagire contemporaneamente con la macchina.

Nel caso di più utenti contemporanei, **il Sistema Operativo deve fornire a ciascuno l'astrazione di un sistema "dedicato"**.

## CLASSIFICAZIONE dei S.O.

Ogni volta che viene lanciato un *programma* da un utente o dal sistema operativo stesso si genera un *processo* che utilizza risorse del sistema.

N.B. Se un programma viene lanciato due volte si generano due processi diversi.

In base al numero di programmi in esecuzione:

- **Mono-programmato (*mono-tasking*)**: si può eseguire *un solo programma* per volta
- **Multi-programmato (*multi-tasking*)**: il S.O. è in grado di portare avanti contemporaneamente l'esecuzione di più programmi (pur usando una sola CPU).

## MONOPROGRAMMAZIONE

Un sistema operativo **monoprogrammato** (tipo il DOS) è capace di gestire un solo programma in esecuzione (un solo processo) alla volta.

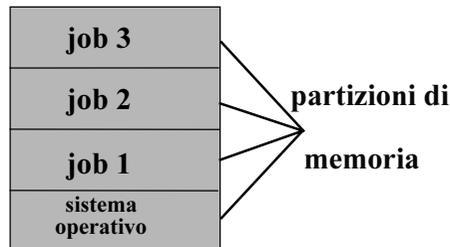
Tutte le risorse del sistema sono allocate a quel processo.

Un secondo processo parte solo quando il primo termina

## MULTIPROGRAMMAZIONE

Un sistema multiprogrammato (tipo UNIX, Windows NT) deve gestire l'allocazione delle risorse a più processi.

Più programmi presenti in **memoria centrale** contemporaneamente



## MULTIPROGRAMMAZIONE(II)

La CPU gestisce un processo alla volta.

- Mentre un programma è in **attesa** del completamento dell'operazione di I/O, la CPU può iniziare l'esecuzione di un altro programma
- **Protezione, gestione della memoria**
- **CPU scheduling** (scelta processi cui assegnare la CPU)
- **Job scheduling** (scelta programmi da caricare in memoria)

Estensione della multiprogrammazione: timesharing

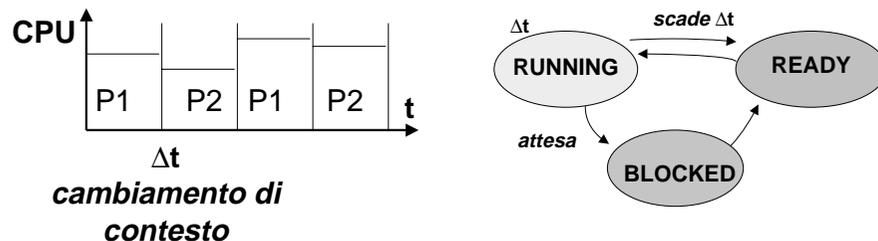


**Il Sistema Operativo deve gestire la suddivisione del tempo** della CPU fra i vari programmi.

## DIVISIONE DI TEMPO (time sharing) NEI SISTEMI MULTITASKING

A ogni programma il S.O. assegna **ciclicamente** un intervallo (*quantum*) di tempo della CPU, fino al suo completamento.

Al termine dell'intervallo (o durante, se il programma inizia un'operazione di I/O) la **CPU** viene assegnata ad un altro prog.



Quando un processo viene interrotto ( $t \leq \Delta t$ ) viene salvato il suo stato che viene ripristinato in modo trasparente all'utente quando la CPU viene riallocata a quel processo

## SISTEMI TIME-SHARING INTERATTIVI

Un **S.O. time-sharing** consente a più utenti di condividere il sistema di calcolo **contemporaneamente** e in modo **interattivo**

**Interattività:** comunicazione diretta tra l'utente e il sistema di calcolo (contrapposto ai sistemi *monoprogrammati*)

Ogni utente ha l'illusione di avere una macchina a sua completa disposizione (Astrazione di più **macchine virtuali**)

Memoria Virtuale: meccanismo di trattamento dello spazio di indirizzamento che permette all'utente di non accorgersi che ci sono altri processi che condividono la memoria.

## SISTEMA OPERATIVO UNIX

Nel corso di sistemi operativi si prenderà in considerazione il sistema operativo **UNIX**

Sistema operativo multiprogrammato a divisione di tempo, interattivo, multiutente.

Varie versioni di Unix: System V e BSD.

Esercitazioni di programmazione di sistema su Workstation SUN del LAB2.

## MEMORIZZAZIONE DI INFORMAZIONI: IL FILE SYSTEM

• Dal punto di vista dell'utente, il file è l'unità logica di memorizzazione delle informazioni.

- Un file è caratterizzato da:
  - un nome (il nome serve per identificarlo senza bisogno di sapere settori tracce ecc..)
  - una dimensione (in byte)
  - degli attributi
  - una data

• Il nome di un file è costituito da:

Identificativo  
univoco



**nomefile.estensione**



Facoltativa:  
identifica il tipo di  
informazioni  
contenute nel file

## TIPI DI FILE

- File di testo: sono file che contengono testo ASCII

### LETTERA.TXT

Spettabile Ditta,  
dal vostro catalogo datato  
Ottobre 1994

- File eseguibili: sono file che contengono "codice macchina"

### MIOPROG.EXE

β\^¥†©÷ÓμÆ@â¢%o|" ‡f  
©™âÆ@μ¢‡f¥^¥@ó\*œ\_μ¢\_  
β\_ ^ÿ@" ‡%o™Ma(¿μ¢%o|" ‡f  
±¤†£÷™â

## TIPI DI FILE

- File (di testo) contenenti programmi

**PROVA.C**: contiene un programma di prova scritto in linguaggio C ("PROVA" è la parte *nomefile* e "C" è la *estensione*).

```
main()
{ int i;
  i = 5;
  printf("Valore di i: %d\n", i);
}
```

- File di dati: contengono informazione strutturata

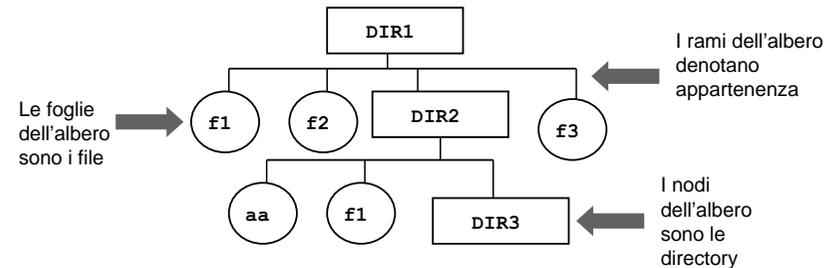
### STIPENDI.DAT

AVANTI LUIGI	1/2/93
L. 2.145.000	
BOSSI PIERO	6/2/93
L. 2.000.000	

## FILE SYSTEM

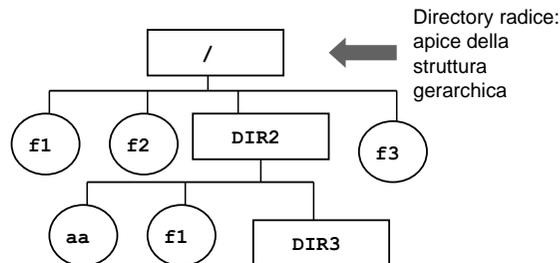
- È l'organizzazione dei file su disco, come vengono strutturati e memorizzati.
  - Ogni unità (disk drive) contiene almeno una *directory*, ossia un "contenitore" in cui memorizzare nome, posizione e altre informazioni di ogni file residente sul disco stesso.
- La *directory* principale (*directory* radice o root) viene creata al momento della *formattazione* del disco. Altre *directory* possono essere create dagli utenti secondo una struttura gerarchica.

## FILE SYSTEM



- Non ci sono limiti teorici al grado di innestamento delle directory.

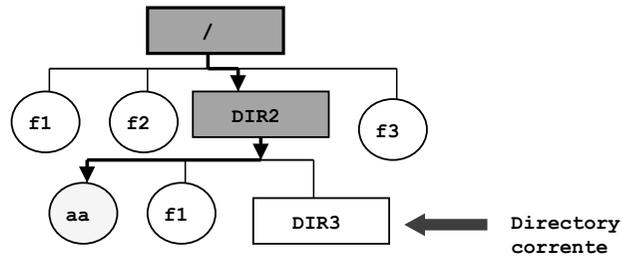
## FILE SYSTEM



## IDENTIFICAZIONE DI FILE E DIRECTORY

- Ogni file o *directory* è identificato da un percorso (*path*) seguito dal suo nome.
- Il percorso è **assoluto** se è riferito alla radice: rappresenta il percorso che si deve seguire a partire dalla radice per giungere alla foglia voluta (file o *directory*).
- Il percorso è **relativo** se è riferito alla *directory corrente* rappresenta il percorso che si deve seguire a partire dalla *directory corrente* per giungere alla foglia voluta.

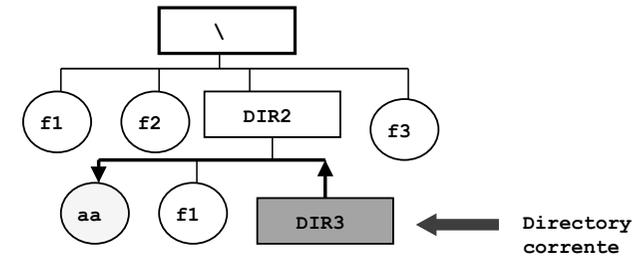
## ESEMPIO: PERCORSO ASSOLUTO



Nome assoluto di aa: /DIR2/aa

- NOTA: la directory corrente non viene considerata nel determinare il percorso assoluto

## ESEMPIO: PERCORSO RELATIVO



Nome Relativo di aa: ../aa

- NOTA: la directory corrente e' il punto di partenza per determinare il percorso relativo

## NOTAZIONI CONVENZIONALI

- In Unix:
  - la *directory* corrente è identificata dal punto (.)
  - la *directory* padre di quella corrente è identificata dal doppio punto (..)
  - la *directory* radice è indicata con '/'
  - il percorso assoluto di un file (o directory) viene indicato specificando nell'ordine il drive su cui si trova il file, il cammino per raggiungerlo inteso come sequenza di directory che si devono percorrere e, infine, il nome del file. Tale sequenza e' separata da simboli '/'
  - il percorso relativo di un file (o directory) viene indicato specificando il cammino per raggiungerlo a partire dalla directory corrente (che non viene indicata).