Storia minima dei sistemi di elaborazione

I sistemi di elaborazione hanno giocato ruoli diversi nel tempo

- **calcolatori**:
in questo ruolo un sistema di elaborazione è soprattutto un dispositivo di calcolo molto veloce (velocità: misurata in MIPS); il concetto centrale è quello di *algoritmo*; gli strumenti principali di accesso alla macchina sono i *linguaggi di programmazione*

- **ordinatori di informazioni**:
in questo ruolo un sistema di elaborazione è "operativo", ovvero capace di archiviare, elaborare, organizzare e accedere grossi insiemi di dati (dimensione dei dati: misurata in bytes); un sistema operativo è un *insieme di servizi*, tra cui spicca quello del *servizio di archiviazione* (o database); lo strumento principale di accesso al sistema ed ai suoi servizi è l'*interfaccia* del sistema operativo

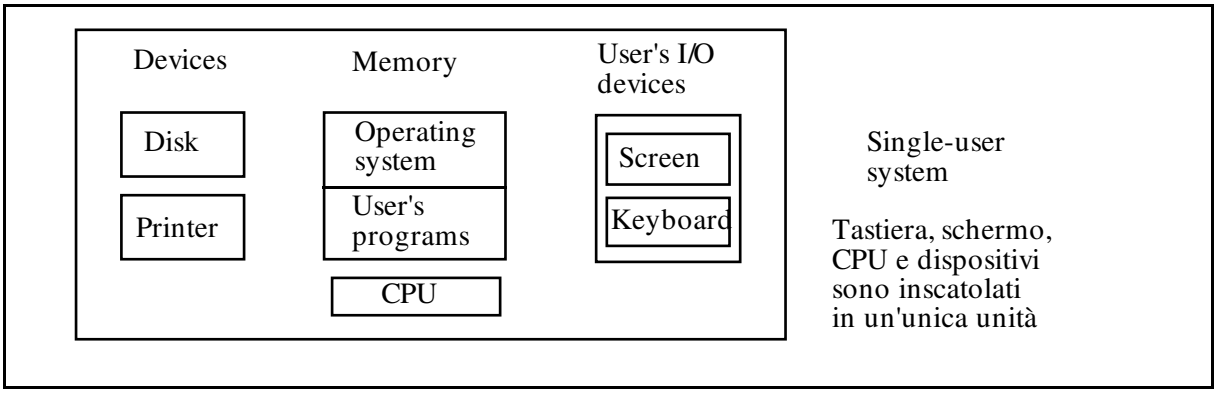
- **strumenti di comunicazione**:
in questo ruolo un sistema di elaborazione è capace di comunicare con altri sistemi collegati in rete, trasportando quantità di dati ad una certa velocità (banda di comunicazione: misurata in baud, o bit/sec); il concetto centrale è quello di servizio di comunicazione (es.: posta elettronica; news; WWW); lo strumento principale di accesso sono i programmi *clienti* dei servizi.

Nota: a noi piace anche ricordare che in molti casi i sistemi di elaborazione sono stati utilizzati come **giocattoli**, e che i programmi di gioco in molti casi sono stati prototipi di applicazioni "utili"

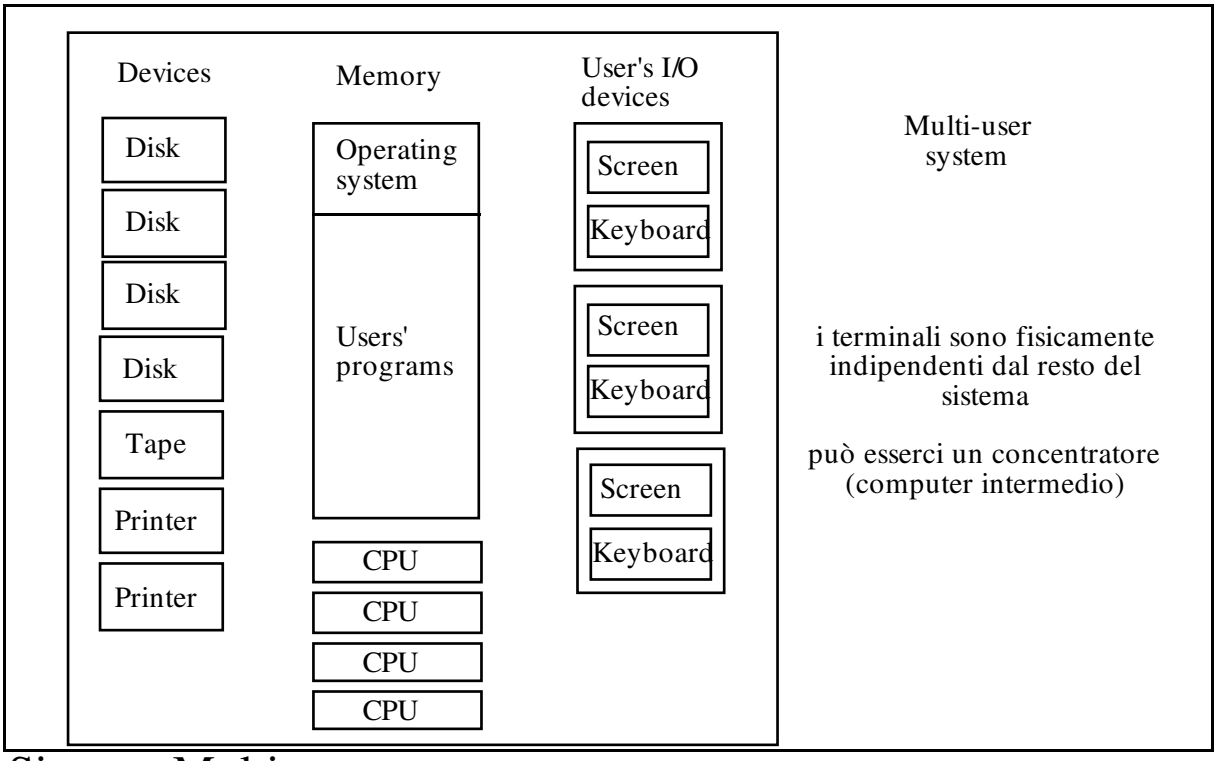
In ogni caso, lo studio dei sistemi di elaborazione si basa sulla nozione di architettura (hardware) e macchina astratta (software)

Architetture dei sistemi di elaborazione

Confronto tra la struttura hardware di un sistema monoutente (monoprocessore) e quella di un sistema multiutente (multiprocessore con memoria condivisa)



Sistema monoprocessore



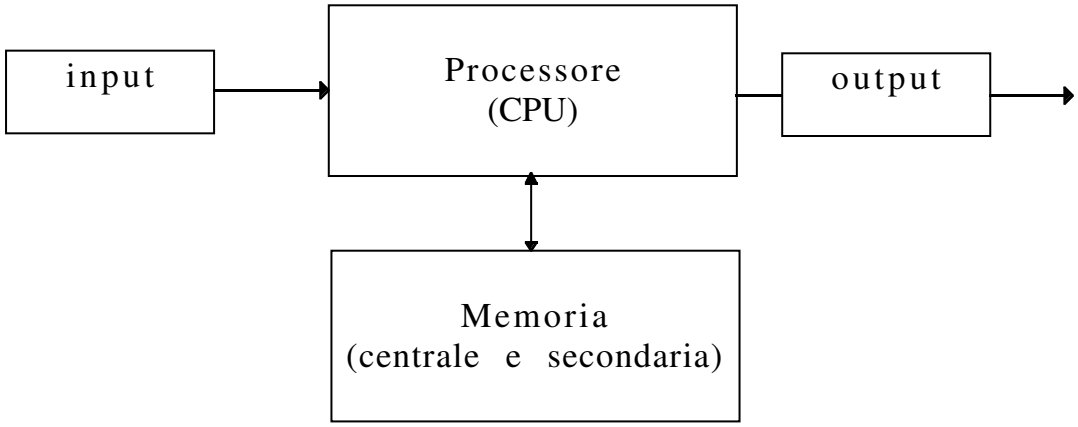
Sistema Multiprocessore

Un modello astratto di sistema di elaborazione

Va notato che l'evoluzione nei campi di applicazione dei sistemi di elaborazione non ha richiesto una corrispondente evoluzione delle funzionalità concretamente offerte dall'hw

I sistemi di elaborazione vengono descritti in astratto da una architettura comprendente 4 componenti hardware:

- processore (CPU)
- memoria (centrale: ROM, RAM; secondaria: dischi, nastri, CDROM)
- dispositivi di ingresso
- dispositivi di uscita



Macchine astratte

Le funzionalità hw (*linguaggio macchina*) sono cambiate poco nel corso degli anni; l'evoluzione è stata invece forte nel caso dei singoli dispositivi, e soprattutto nel calo dei costi, grazie alle economie di produzione su larga scala

Sono invece molto evolute le funzionalità delle *macchine astratte* realizzate dal sw di un sistema operativo

- interfaccia
- servizi
- applicazioni

Una *macchina astratta* è definita da un *linguaggio*, ovvero da una sintassi (comandi della macchina astratta) e da una semantica "operazionale" (operazioni che la macchina esegue su uno stato astratto in risposta ai comandi)

Esempi di macchina astratta:
- un editor di testo, definito dai comandi che agiscono sull'astrazione testuale
- un cliente di posta elettronica, definito dai comandi che agiscono sull'astrazione di casella postale (insieme di messaggi) di un utente

Le macchine astratte, esattamente come quelle concrete, sono combinabili in macchine più complesse (architetture software). Es.: un programma editor che può diventare un browser WWW

Il s.o. è una gerarchia di macchine astratte

Un *sistema operativo* è un programma che controlla e amministra le risorse di calcolo, memorizzazione e comunicazione di un sistema di elaborazione, rendendole accessibili agli utenti ed ai loro programmi

Il s.o. integra e rende più facilmente usabili una quantità di risorse di natura diversa: processore, memoria, dispositivi di I/O, rete, ecc.

Il s.o. può essere caratterizzato come **gerarchia di macchine astratte**, strutturate in una *architettura di sistema*

L'utente ha a disposizione una serie di linguaggi di natura diversa, che gli permettono di controllare in vari modi le risorse del sistema

- Interfaccia Utente
  - Testuale
  - Grafica a finestre
- Ambiente di programmazione
  - Insieme di strumenti: editor, compilatore, debugger, …
  - Supporti: librerie, database di progetto, …
- Linguaggi di sistema
  - Linguaggio di programmazione delle applicazioni
  - Linguaggio di costruzione del s.o. (es.: il C)
  - Linguaggio di configurazione (es.: makefile)
  - Linguaggio di interfaccia utente
  - Linguaggio dei servizi di sistema
- Meccanismi primitivi di gestione di risorse
  - Il nucleo e la gestione del processore
  - La gestione della memoria centrale
  - Connessione di rete
  - File system (centralizzato, distribuito)

Servizi dei sistemi di elaborazione

Interfaccia amichevole

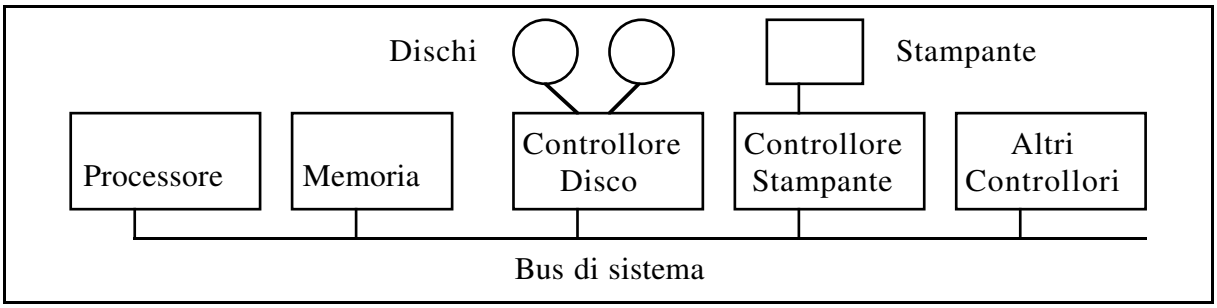
Nei sistemi più primitivi (DOS) l'interfaccia utente è normalmente di tipo testuale; in quelli moderni se il terminale è grafico si usano le interfacce grafiche e le loro librerie (Windows, X-Window, SunTools)

Time-sharing

La configurazione di un sistema *time-sharing* comprende, oltre all'unità centrale di elaborazione, più terminali interattivi e più dispositivi di memoria secondaria (generalmente dischi)
Nei sistemi *time-sharing,* la multiprogrammazione adotta una politica di gestione del processore che limita l'esecuzione di ogni programma a un *quanto di tempo,* di valore relativamente piccolo

Condivisione di risorse e memorizzazione permanente

Tra i servizi erogati, sono importanti quelli per la condivisione di risorse di I/O (es. stampanti) e per la *gestione degli archivi* (es. su disco): l'utente identifica gli archivi con nomi simbolici e dispone di comandi per la creazione, il reperimento, la lettura, la scrittura e l'eliminazione. Un altro importante servizio è quello che consente di scrivere o di modificare gli archivi dal terminale *(editor).*



Architetture dei sistemi più diffusi

Mainframe

I mainframe sono grandi sistemi molto più costosi di una workstation. Offrono elevata potenza di calcolo, ampie risorse di memorizzazione, parecchi terminali interattivi e di solito funzionano in time-sharing.

Workstation e calcolatori personali

I calcolatori personali offrono all'utente un file system gerarchico ed un sistema operativo monoprogrammato o al più multiprogrammato senza multitasking; solo di recente anche il multitasking comincia ad essere disponibile; l'interfaccia utente è grafica (es.: Mac, Windows)

Network computer (e WebTV)

La diffusione della rete permette di costruire piccoli computer a basso costo, con processore, memoria centrale e disco limitati, integrati in dispositivi TV connessi a Internet. Gli utenti trovano in rete le risorse ed i programmi di cui hanno bisogno.

Sistemi operativi con vincoli di tempo reale

Sono sistemi operativi utilizzati in controllo di processi industriali o in monitoraggio di attività critiche in cui il rispetto dei tempi di esecuzione è essenziale

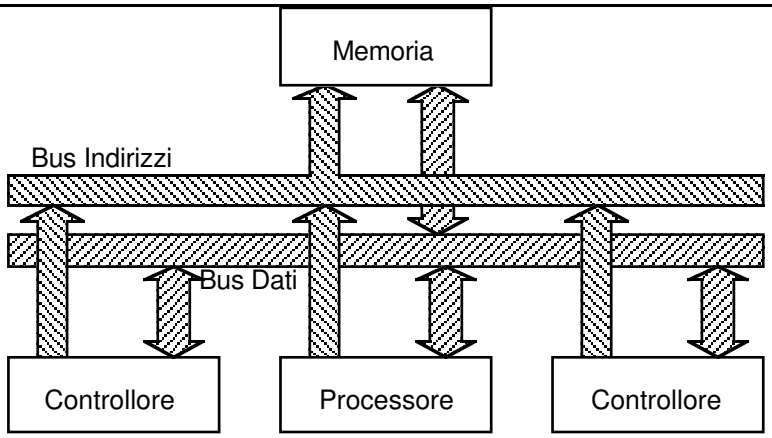
Sistemi operativi distribuiti

Sono sistemi operativi che controllano una rete o comunque un insieme di processori, offrendo agli utenti un accesso *trasparente* a tutti i servizi e risorse

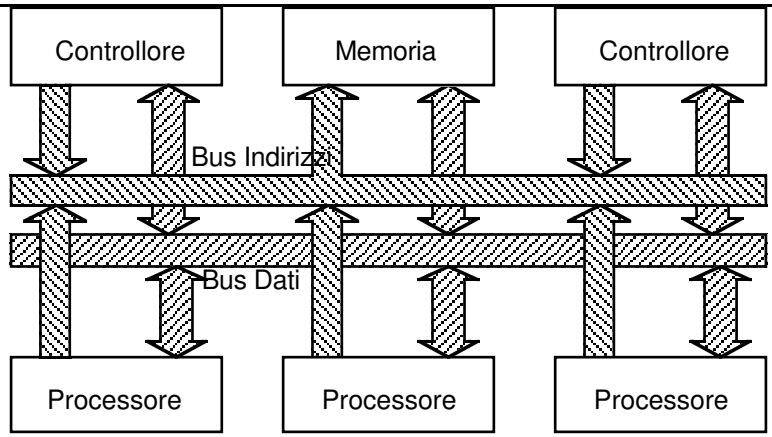
Struttura di un sistema di elaborazione

• Unità centrale di elaborazione (CPU: Central Processing Unit).
• Dispositivi periferici: di memoria secondaria; di I/O

Il trasferimento di dati tra processore, memoria principale e dispositivi avviene attraverso il *bus* (struttura di comunicazione condivisa)



Flusso degli indirizzi e dei dati negli accessi alla memoria in un sistema uniprocessore



Sistema multiprocessore

Memoria principale (Main Storage, MS)

- Vettore di *celle,* ciascuna identificata dal suo *indirizzo.*

- Ogni cella può contenere una unità di informazione
  (*byte* ovvero *parola)*

La capacità di memoria centrale si misura in Megabyte (Mb); tipiche dimensioni per PC moderni sono 16, 32, 64 Megabytes

- Moduli di memoria
  ***R**ead-**W**rite **M**emory (RWM),*
  ***R**ead-**O**nly **M**emory (ROM).*
- Accesso "casuale" o ***R**andom **A**ccess **M**emory (RAM).*

- Accessi a MS eseguiti da:
  - processore (o dai processori)
  - controllori dei dispositivi con DMA

- Conflitti per l'accesso al bus risolti dall'*arbitro:*
  riceve richiesta (segnale) *RBus;*
  risponde con abilitazione (segnale) *ABus*

La comunicazione del processore con i dispositivi può avvenire anche attraverso *porte* di ingresso/uscita che fanno parte della memoria principale

Il processore (o CPU)

Un processore è in realtà composto da parecchi componenti

• **Unità di controllo** Decodifica i codici delle istruzioni

• **Unità Aritmetico-Logica** (Arithmetic-Logic Unit - ALU)

• **Temporizzatore** (Orologio di macchina)

La velocità del processore si misura in MIPS o in Mhz.
La capacità dei registri (es: 16, 32 o 64 bit) determina SIA la quantità di informazione elaborabile in parallelo SIA la dimensione della memoria centrale indirizzabile

• **Registri generali**

I registri generali (accumulatori) servono a gestire gli operandi e i risultati delle istruzioni di macchina; alcuni registri generali sono dedicati a speciali funzioni e alcune istruzioni li riferiscono implicitamente.
Esempio: registri indice, puntatore della pila.

• **Registri speciali**

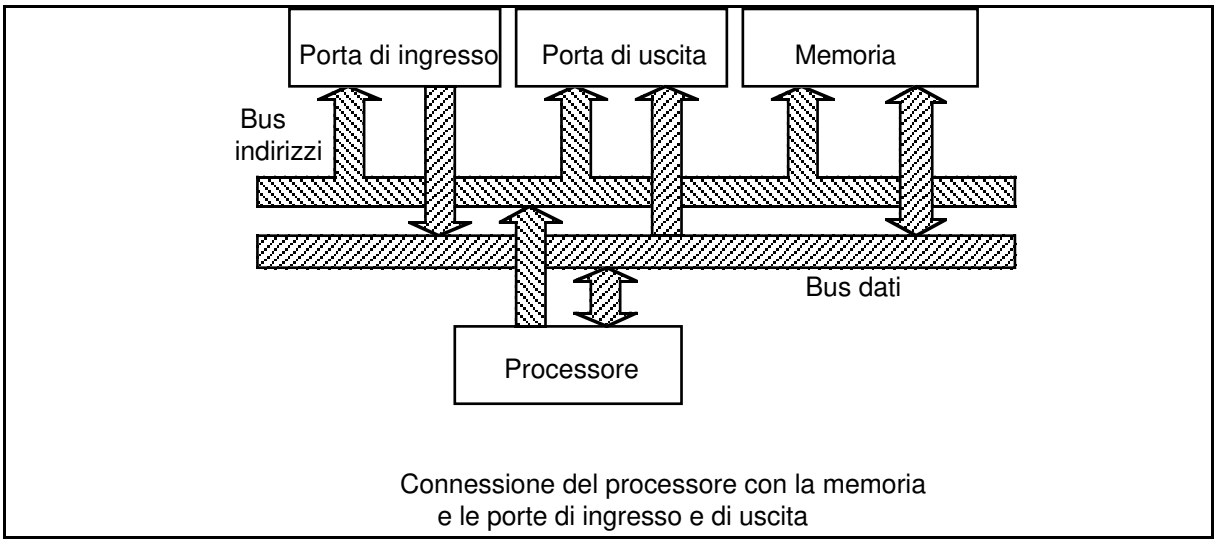
Non sono esplicitamente accessibili con istruzioni di macchina, salvo le istruzioni privilegiate. Esempi:

- Contatore di programma o Program Counter (PC)
- Registro di istruzione o Instruction Register (IR)
- Registri per la generazione dell'indirizzo (registri di segmento)
- Registri per la traduzione degli indirizzi logici in indirizzi fisici
(registri base e registri limite;
oppure chip specializzato Memory Management Unit (MMU))

Il processore: interazioni coi dispositivi

Comunicazione tra processore e dispositivi periferici
- attraverso le *porte* di ingresso o di uscita.
- con accesso diretto alla memoria (DMA)

Porte di ingresso/uscita
- Una porta di ingresso è un registro che può essere:
  scritta da un dispositivo periferico
  letto dal processore attraverso il bus.
- Una porta di uscita è un registro che può essere:
  scritto dal processore attraverso il bus
  letto da un dispositivo periferico.
- Capacità di ogni porta: una unità di informazione (byte o parola).



Connessione del processore con la memoria e le porte di ingresso e di uscita

- *I/O-mapped I/O:* Ad ogni porta è associato un indirizzo, definito in uno spazio distinto da quello della memoria principale.

- *memory-mapped I/O* (alternativa): gli indirizzi delle porte sono definiti in un sottoinsieme dello spazio degli indirizzi della memoria principale (non utilizzabili per riferire celle di MS).

I dispositivi periferici

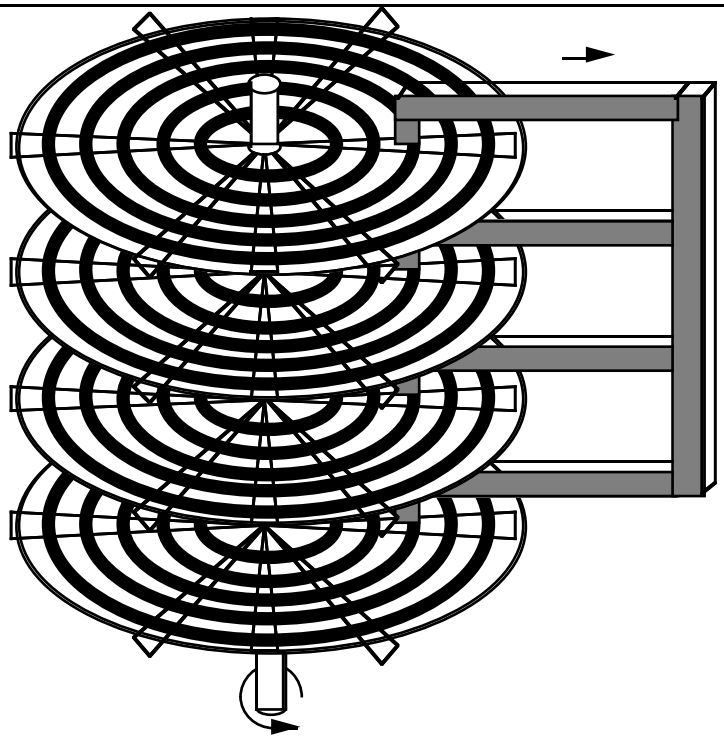
Classificazione funzionale:
- Dispositivi di ingresso
  esempio: tastiera, mouse, joystick, ecc.
- Dispositivi di uscita
  esempio: video, altoparlante, stampante, ecc.
- Dispositivi di memoria secondaria o ***B**acking **S**tore (BS)*
  esempio: dischi rigidi, nastri, floppy disk, compact disk
- Temporizzatori (timers)
  fisicamente fanno parte del processore.

Classificazione in base all'interazione:
- Dispositivi a **caratteri**
  Comprendono dispositivi di ingresso e di uscita
  Ogni operazione (lettura o di scrittura) trasferisce una singola unità di informazione
  Trasferimento avviene attraverso una porta
- Dispositivi a **blocchi**
  Tipicamente dispositivi di memoria secondaria:
  ogni operazione trasferisce un *blocco* con sistema di *DMA,* sotto il controllo di un *controllore di DMA*

DISPOSITIVI DI MEMORIA SECONDARIA

Differenze rispetto alla memoria principale:
- tempo di accesso superiore: ordine di 10 msec
- costo per unità di informazione molto minore, quindi capacità molto maggiori rispetto alla memoria principale
- memorizzazione persistente dei dati
- accesso sequenziale (nastri) o diretto (dischi)

Dischi Indirizzo: *(cilindro, faccia, settore)*
Tempo di accesso: tempo di ricerca *(seek)* + tempo di latenza + tempo di trasferimento.



Disco magnetico a teste mobili

Dischi "rigidi": distanza piccolissima delle teste dalle facce; disco in movimento costante.
Dischetti (floppy-disks): manca un controllo accurato della distanza delle teste dalle facce; usure del materiale magnetico, disco arrestato quando non è in uso.
Dischi ottici (CD ROM): lettura laser

Floppy, Hard e Compact Disk

• Floppy disk: unità con piccola capacità di memorizzazione
~1.4 Mbytes nel caso di fd ad alta densità

• Hard disk "fissi": grandi capacità di memorizzazione
~1-20 Gbytes

• Hard disk "removibili": da 100 Mb a 1 Gb

• Compact disk DA (digital audio): standard per CD musicali (max 74' digitali) definito ds Philips e Sony nel 1982

• CD ROM: nessuno standard, di fatto definito da Philips-Sony nel 1985. Esisono una serie di "colored books":
- *Red book*: definisce il CD DA come 650 Mb su traccia continua di blocchi da 2.532 byte, da leggersi a 75blocchi/sec (pari a 170.400 bps ~150Kbps = 1x)
- *Yellow book*: CD DA + correzione errori, per supportare CDROM. Ogni blocco contiene 2048 bytes di dati e 280 byte di correzione errori
- *Green book*: estensione di Yellow per CD Interactive (standard proprietario Philips)
- *Orange Book*: CD R (Recordable)
- *Blue book*: CD Extra, cioè audio+dati (1996)
- CD Rw (Rewritable) (1997)

• Digital video disk (DVD): è molto più denso del CD DA, contiene infatti 4,7 Gb per lato. La superficie può avere due livelli, allora contiene 8,5 Gb per lato

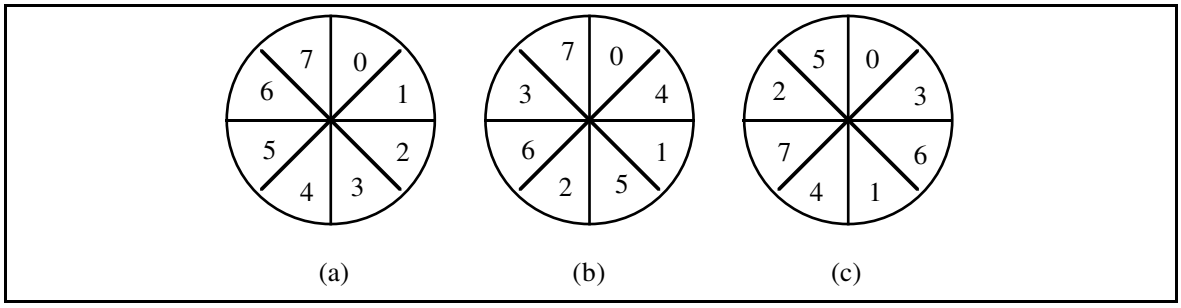
Nota sul costo dell'unità di memorizzazione:
memoria centrale: ~10.000 lire/Mb
floppy disk: ~700 lire/Mb
CD R: ~15 lire/Mb
hard disk removibile (Zip) 100 Mb: ~300 lire/Mb
hard disk removibile (magneto-ottico) 230 Mb: ~110lire/Mb

Gestione del disco dal punto di vista hardware

Mentre i dati vengono trasferiti dal controllore del disco alla memoria principale, il settore successivo starà passando sotto la testina di lettura ed i relativi bit staranno arrivando al controllore e verranno persi

Se il tempo di trasferimento di un blocco dal controllore alla memoria è maggiore del tempo che occorre per leggere un blocco da disco, potrebbe essere conveniente leggere un blocco e poi saltare 2 (o più) blocchi

Il fatto di saltare blocchi senza leggerli per dare il tempo al controllore di trasferire i dati in memoria viene chiamato *interleaving*
Quando il disco viene formattato, i blocchi sono numerati tenendo conto del fattore di interleaving

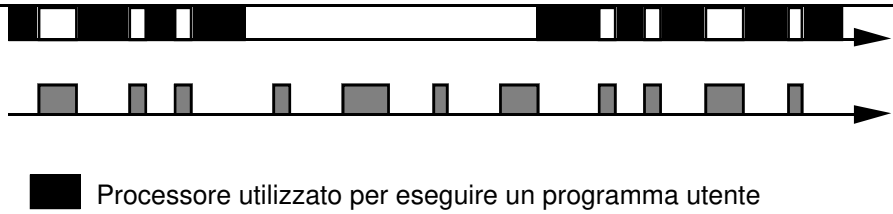


In (a) vediamo un disco con 8 settori per traccia senza interleaving
In (b) vediamo lo stesso disco con interleaving singolo
In (c) si vede il doppio interleaving.

In questo modo si permette alla testina di leggere i blocchi numerati consecutivamente mantenendo tuttavia la massima velocità di rotazione del disco

Parallelismo delle funzioni hardware

Molte delle funzioni di un sistema operativo sono svolte da programmi attivati da *interruzioni*; grazie al supporto del sistema di *DMA,* ogni attivazione impegna il processore per breve tempo



Utilizzo del processore in un sistema operativo a lotti uniprogrammato

Quando il programma richiede una lettura o una scrittura e non può continuare prima che queste operazioni siano state completate, il processore è utilizzato solo per eseguire i programmi attivati dalle interruzioni; per recuperare i tempi morti è opportuno virtualizzare i dispositivi

(Ai sistemi operativi con dispositivi virtuali si deve l'introduzione del concetto di *indipendenza* delle funzioni e delle politiche dai dispositivi e dai meccanismi utilizzati per realizzarle)

Esempio: dispositivo lettore virtuale
L'ordine con cui procede l'esecuzione dei lavori è influenzato dalla politica adottata per creare e assegnare al lettore virtuale i lotti di ingresso, ma non dall'ordine con il quale le immagini delle schede sono registrate sul disco

Gli archivi vengono letti dal lettore virtuale nel formato specifico per i lotti; le eventuali conversioni dal formato delle schede e dei pacchi di schede compete solo ai programmi che controllano i dispositivi fisici

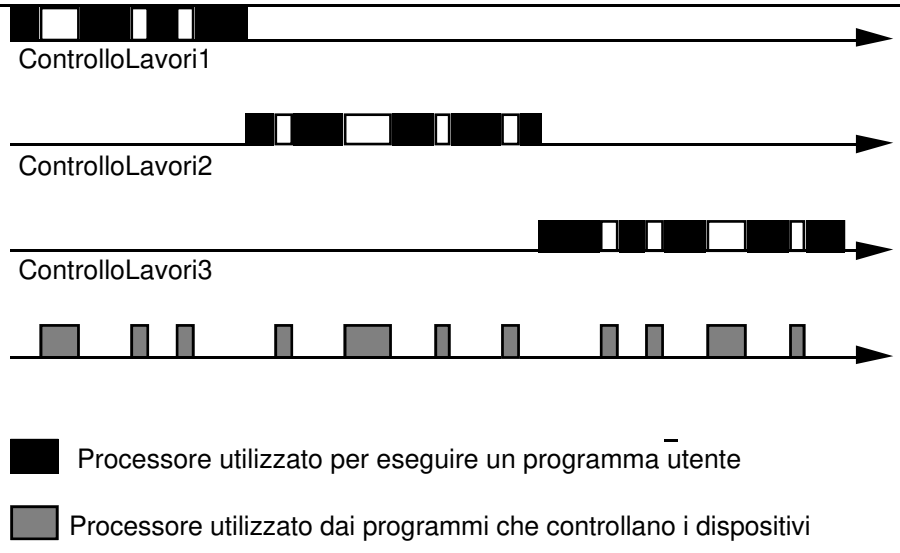
Multiprogrammazione

Il tasso di utilizzazione del processore nel caso uniprogrammato dipende dalla frequenza di letture o scritture invocate dal programma

Un sistema operativo multiprogrammato include *n* distinte attività che svolgono la funzione di *ControlloEsecuzioneLavori*

Queste attività si alternano sul processore (multitasking): quando la generica attività *ControlloEsecuzioneLavoriI* si sospende per una lettura o una scrittura, il processore viene utilizzato per un'altra attività *ControlloEsecuzioneLavoriJ* che sia pronta ad avanzare

Le attività di *ControlloEsecuzioneLavori* non avanzano con continuità, ma sono interrotte per brevi intervalli di tempo da quelle che controllano i vari dispositivi



Utilizzo del processore in un sistema operativo a lotti multiprogrammato

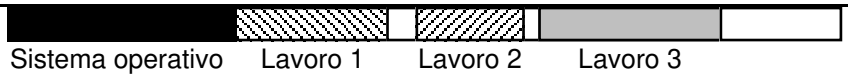
Gestione delle risorse in multiprogrammazione

In caso di multitasking la funzione di gestione del processore sceglie di volta in volta l'attività da fare avanzare tra quelle pronte

Le attività che controllano i dispositivi hanno la precedenza su quelle di *ControlloEsecuzioneLavori*. Anche tra le attività che controllano i dispositivi esistono generalmente diversi livelli di priorità, a seconda del tipo di dispositivo controllato

L'avanzamento "contemporaneo" di più attività è possibile perché la memoria è ripartita tra i rispettivi programmi. Questo compito spetta alla funzione di *gestione della memoria* del sistema operativo

Un esempio di ripartizione spaziale della memoria è quello che risulta dal caricamento con partizioni variabili, dove la memoria è attribuita alle *n* funzioni di *ControlloEsecuzioneLavori*



Gestione della memoria con partizioni variabili
in un sistema operativo multiprogrammato

Servizi e funzioni del sistema operativo

Servizi

Archiviazione di dati e programmi
- librerie
- database di progetto

Esecuzione di programmi
- ambiente di programmazione (editor, compilatore)
- esecutori (interprete, debugger)

Ingresso e uscita
- dispositivi di visualizzazione e stampa

Comunicazione e gestione dell'informazione remota
- posta elettronica
- remote login
- remote file transfer
- news
- WWW

Funzioni

Allocazione di risorse (processore, memoria, dispositivi)
Ottimizzazione della gestione (politiche)
Soluzione dei conflitti (sincronizzazione)

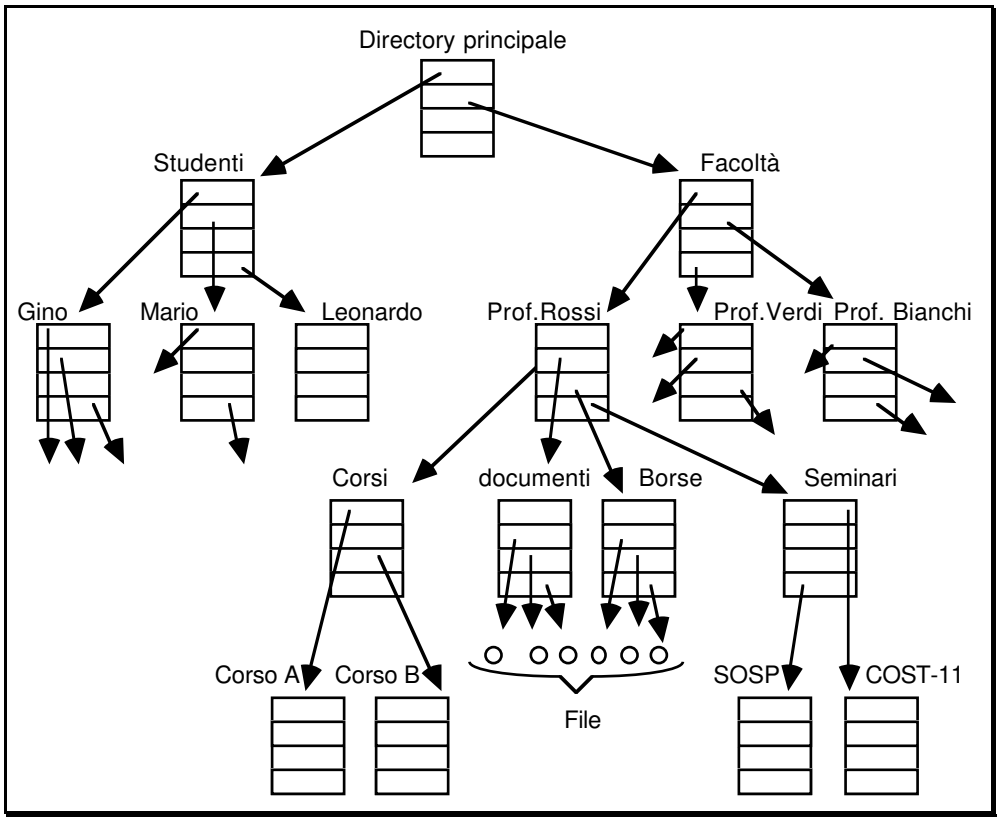
Servizi di archiviazione

Un *file* è uno "spazio di archiviazione" identificato da un nome, in cui si memorizzano caratteri (byte) che rappresentano dati oppure programmi

L'insieme dei file, detto *file system*, ha una struttura ad albero (gerarchica); i nodi dell'albero si chiamano *directory*

Unix ammette 3 tipi di file:

Normale (sequenza di byte): una foglia dell'albero
Directory (insieme di file): un nodo interno dell'albero
File speciale: rappresenta un dispositivo di I/O



Servizi di gestione/esecuzione di programmi

Il sistema operativo Unix(Linux) permette tra l'altro:

- Manipolazione di testi
  - editor `vi, emacs`
  - manipolazione di file: `sed`, `perl`

- L'ambiente C
  - Il compilatore (il linker, l'ottimizzatore)
  - Il debugger simbolico
  - `make`, SCCS, RCS

- Formattazione di testi
  - `nroff, troff`
  - `TeX, LaTeX`

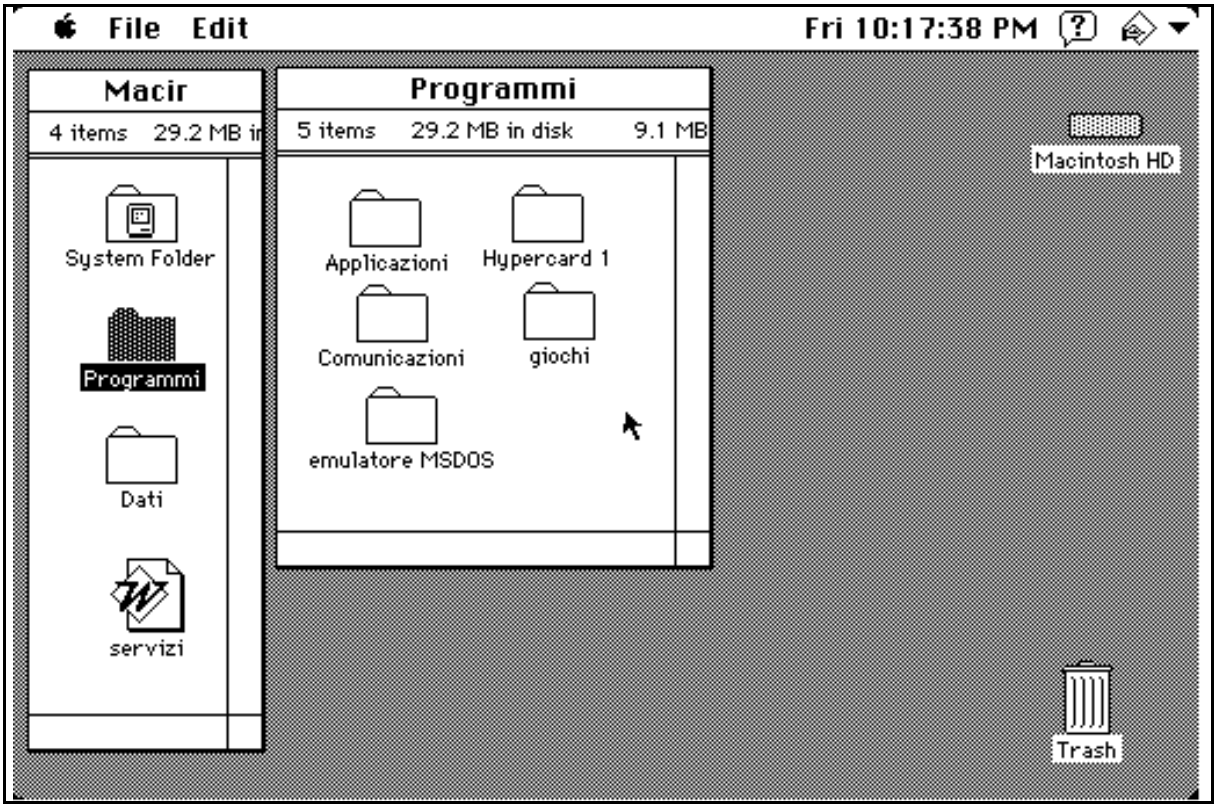
- Archiviazione, codifica e compressione
  - `tar, compress, uucode, gzip`

Interprete dei comandi (shell)

I comandi che l'utente trasmette attraverso il terminale sono riconosciuti dall'*interprete dei comandi;* è un programma che ripete un ciclo all'interno del quale attende ed esegue un nuovo comando

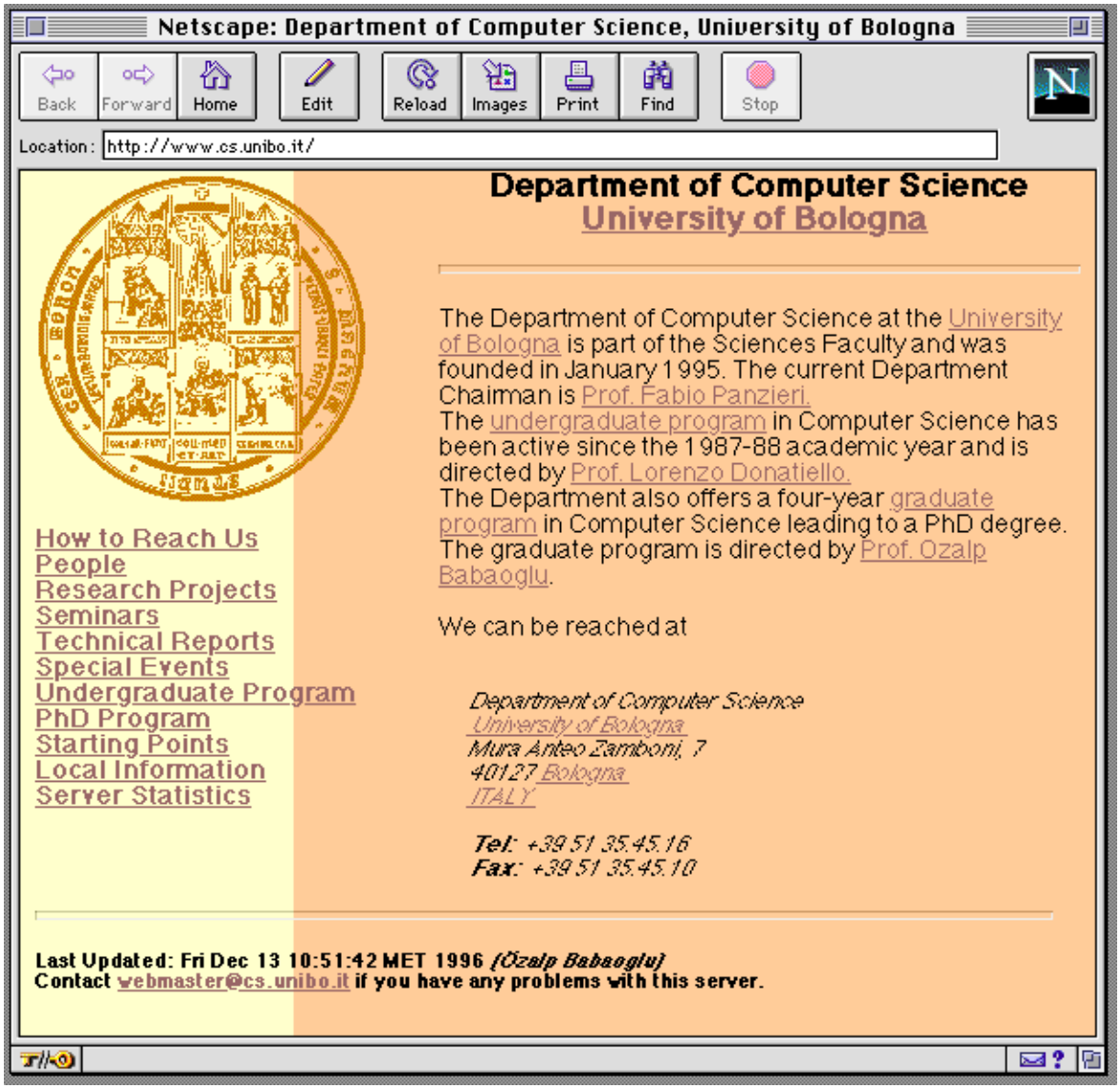
In alcuni sistemi l'interprete dei comandi è considerato una funzione del sistema operativo, alla stessa stregua della gestione della memoria o della gestione degli archivi. In altri casi è concepito come un processo (agente) interattivo

L'interfaccia verso l'utente può essere molto diversa nei vari sistemi operativi; lo schema più semplice è quello in cui il sistema operativo sollecita un comando testuale al terminale; trovano crescente diffusione le cosiddette *interfacce visuali,* nelle quali il sistema facilita l'utente proponendo sullo schermo il *menu* dei comandi che sono selezionabili mediante dispositivi di puntamento sullo schermo, come ad esempio il *mouse*



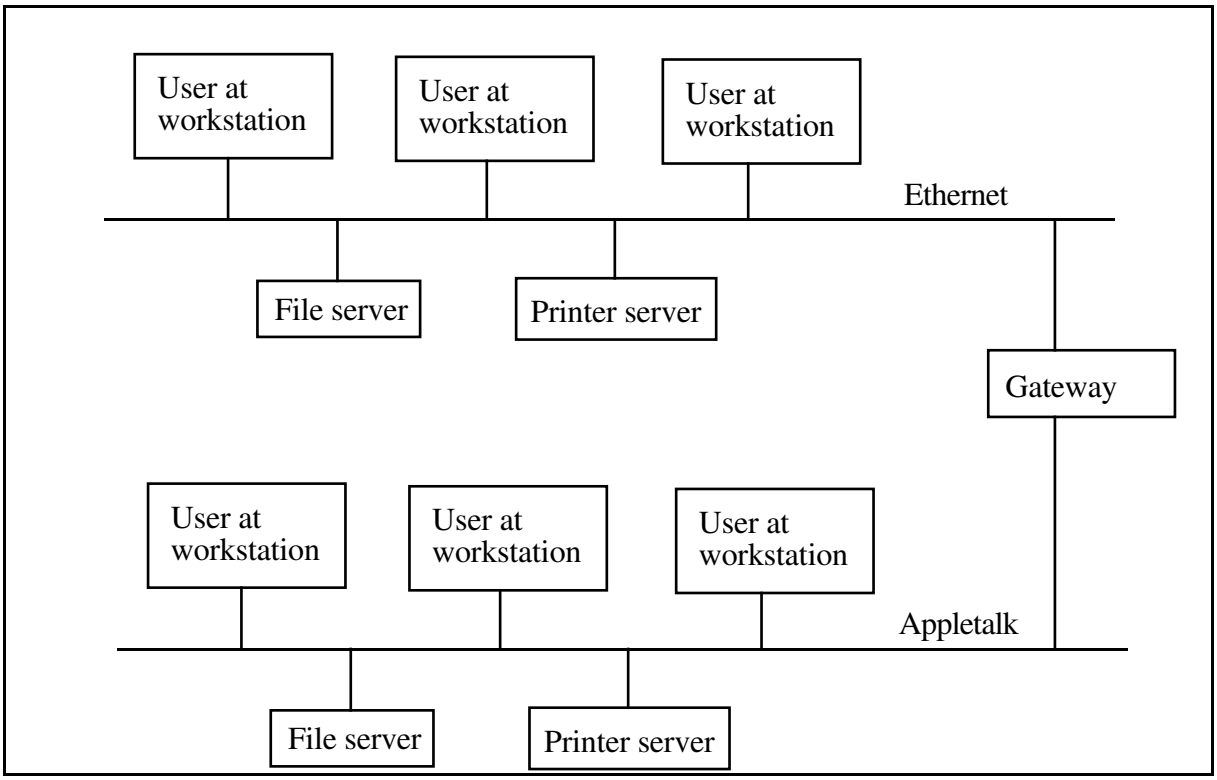
Browser

Un browser è al momento un'applicazione che permette di collegarsi a Internet via un servizio WWW; sembra che sia destinata a diventare la principale metafora di riferimento come interfaccia di sistema operativo (vedi cause Microsoft-Netscape e Microsoft-Sun)



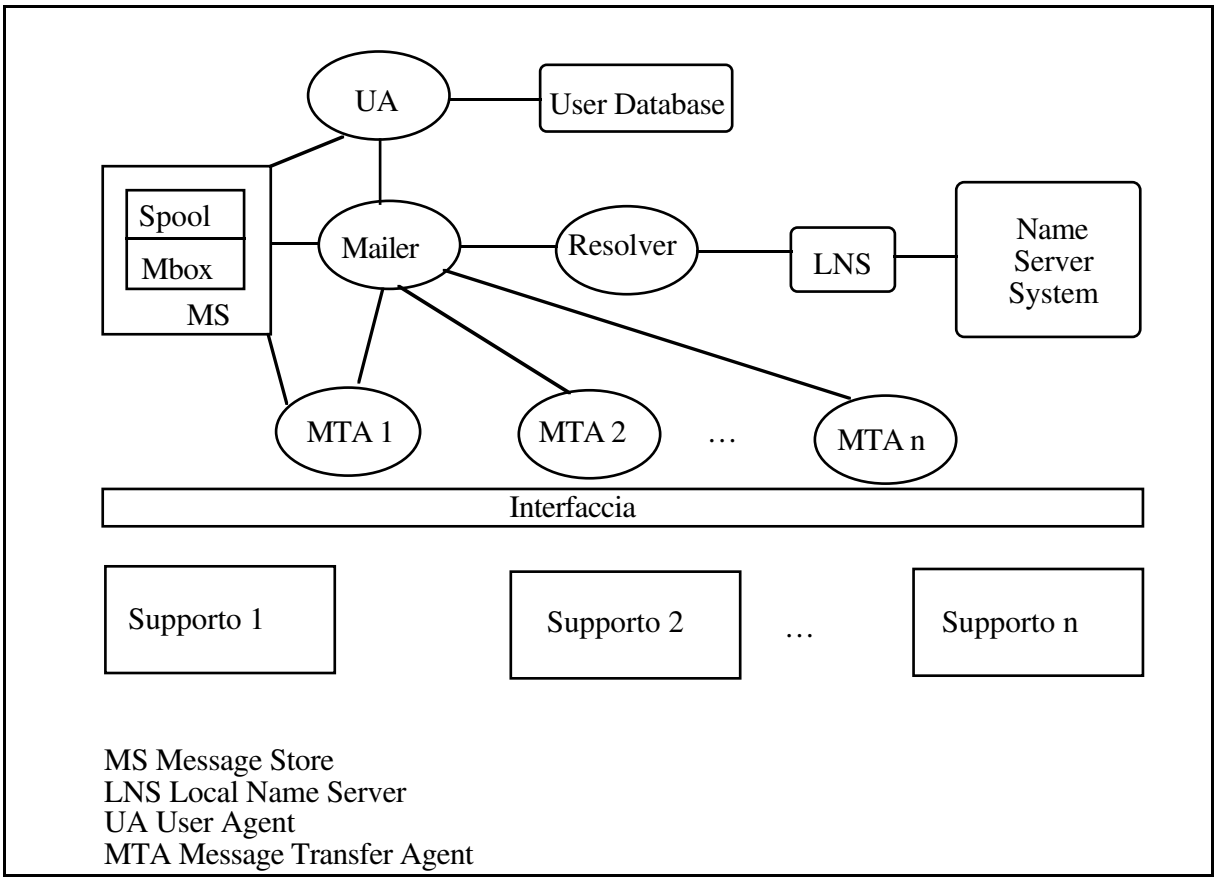
Servizi di comunicazione

Le moderne tecnologie telematiche permettono di connettere in rete locale centinaia o addirittura migliaia di computer che condividono servizi di sistema



Posta elettronica

La posta elettronica è forse l'applicazione di rete oggi più diffusa e utile; esistono moltissime realizzazioni, che si possono riassumere in un modello generale come segue



Un messaggio è in sostanza un record

• lo *user agent* UA è un programma che interagisce con l'utente nella composizione del msg; riempie alcuni campi in modo automatico (indirizzo mittente, data composizione, Reply-to, riservatezza, priorità, sender, ecc.).
• il *mailer* usa il campo indirizzo del destinatario e consegna il msg se il destinatario è locale
• il *resolver* risolve gli indirizzi non locali interrogando il name server locale e in subordine il servizio di name server distribuito
• il *message transfer agent* MTA provvede al trasporto del messaggio, dialogando via rete con un MTA remoto (più MTA possono essere coinvolti nel routing)

Indirizzi su Internet

Gli indirizzi di internet si dividono in:

indirizzi assoluti di host:
`130.136.2.6` (sempre 4 byte)

indirizzi simbolici di host:
`susanna.cs.unibo.it` (lunghi a piacere)

indirizzi di utenti: `ciancarini@cs.unibo.it`

L'hardware di rete è capace di usare solamente gli indirizzi assoluti e quindi è necessario un meccanismo di conversione delle stringhe ASCII in indirizzi di rete

Ai tempi di ARPANET esisteva un unico file, `hosts.txt`, che elencava tutti gli host e i loro indirizzi IP. Ogni notte tutti gli host della rete se lo copiavano scaricandolo dal sito in cui era mantenuto

Quando la rete crebbe fino a comprendere migliaia di macchine venne inventato il DNS (Domain Name System).

Quando un programma deve trasformare un nome in un indirizzo IP chiama una procedura di libreria detta risolutrice (resolver), passandole il nome come parametro di ingresso. Il resolver manda un msg ad un server DNS locale, che cerca il nome e restituisce l'indirizzo al resolver, che a sua volta lo trasmette al programma chiamante

Usando tale indirizzo IP il programma può aprire una connessione TCP con la destinazione

Lo spazio dei nomi su Internet

Internet è suddivisa in centinaia di *domini radice*; ciascuno si divide in sottodomini, a loro volta partizionati. Tutto questo si rappresenta graficamente con un albero

Il nome di un dominio è composto dal cammino inverso dalla foglia fino alla radice (anonima). Un dominio foglia può contenere un singolo host oppure migliaia, e può rappresentare un'intera società

I domini radice possono essere generici o nazionali. I domini generici sono `com` (commerciale), `edu` (istituzioni educative), `gov` (il governo federale USA), `int` (alcune organizzazioni internazionali), `mil` (le forze armate USA), `net` (fornitori di servizi di rete), e `org` (organizzazioni senza scopo di lucro). I domini nazionali sono uno per nazione, e sono definiti nel documento ISO 3166.

In linea di principio i domini possono trovare posto nell'albero in più modi. Ad esempio, `cs.yale.edu` potrebbe essere inserito in alternativa nel dominio nazionale `us`, come `cs.yale.ct.us`.

Nessuna regola proibisce di registrarsi sotto due domini radice, ma, poiché la cosa potrebbe generare confusione, poche organizzazioni si avvalgono di questa possibilità.

Ciascun dominio controlla l'allocazione dei suoi sottodomini. Ad esempio, il Giappone ha i domini `ac.jp` e `co.jp` che ricalcano `edu` e `com`. L'Italia non fa distinzioni e mette tutte le organizzazioni sotto `it`. Quelli che seguono sono tre dipartimenti di informatica:

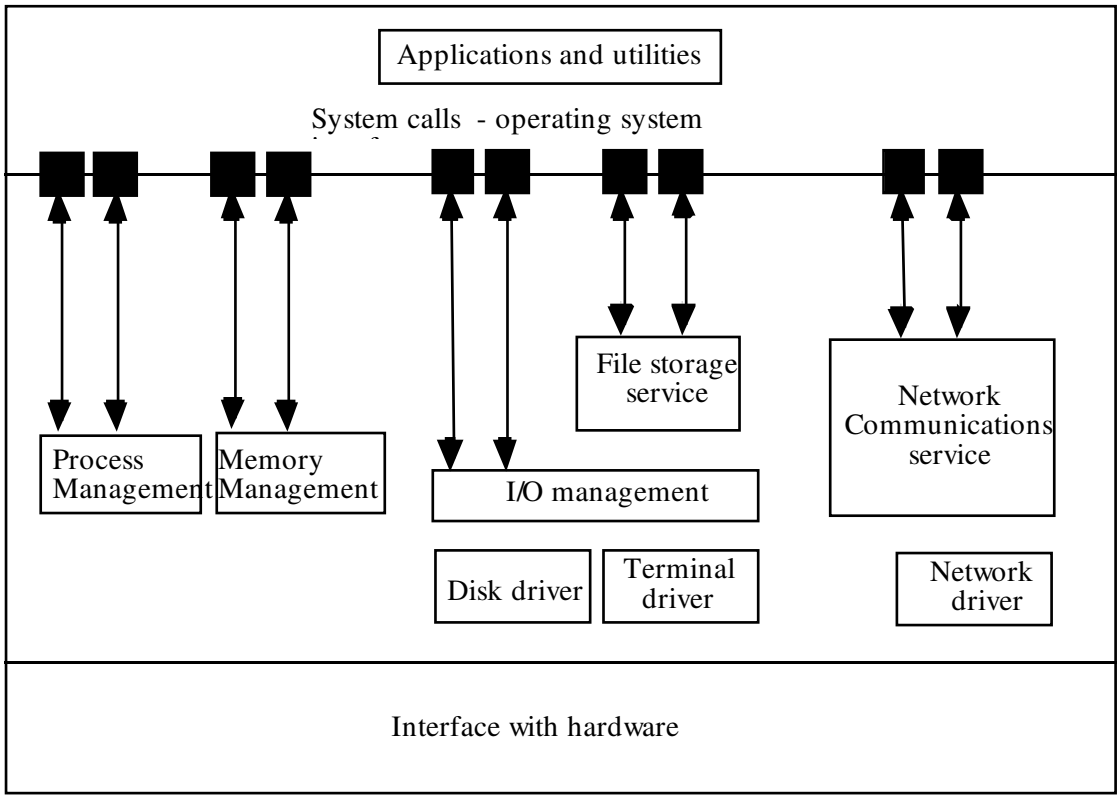
1. `cs.yale.edu` (Yale University negli USA)
2. `cs.vu.nl` (Vrije Universitet in Olanda)
3. `cs.keio.ac.jp` (Keio University in Giappone)

Architettura di un sistema operativo

Un sistema operativo è una macchina astratta complessa

Viene di solito descritta definendo un insieme di agenti chiamati *processi*

- processi di interfaccia (shell)
- processi delle applicazioni
- processi di servizio (serventi)



La nozione di processo

Un sistema operativo è un programma complesso; per capirne la struttura e le attività si introduce la nozione di *processo:*

Un *processo*, o *agente*, è un'attività controllata da un programma che si svolge su un processore

Nota: il sistema operativo NON è un processo; viene definito invece come il sw necessario a realizzare l'astrazione di processo, più alcuni processi permanenti (*server*, o *demoni*)

Un processo *avanza* utilizzando un processore (in modo esclusivo); l'avanzamento non è continuo (a causa di interruzioni, sospensioni e prerilasci)

Possiamo rappresentare un processo mediante il concetto di *traccia di stati*

Stato di un processo: rappresentato come *P= (IM, IP, S)*

IM: immagine in memoria,
- informazione costituita da programma e dati
- modificata dal processo che avanza (solo i dati)

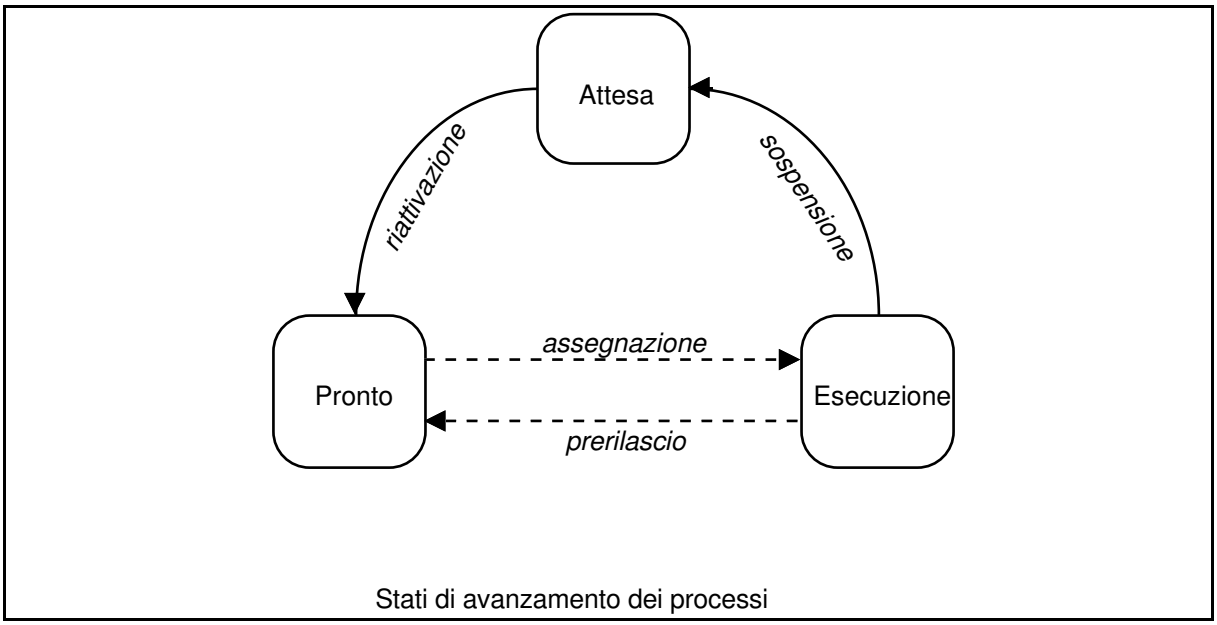
IP: immagine nel processore
- informazioni sul processo mantenute nei registri del processore
- modificata dal processo che avanza

S : *stato di avanzamento* (o brevemente *stato).*
- assume i valori di *attesa,* di *pronto* e di *esecuzione.*

- Stato di *esecuzione*: il processo dispone del processore e avanza nell'esecuzione del suo programma.
- Stati di *pronto* e di *attesa*: il processo non dispone del processore e non avanza

Stati e transizioni di stato

Ogni processo durante la sua "vita" alterna gli stati di attesa, di pronto e di esecuzione passando attraverso *transizioni di stato*



Stati di avanzamento dei processi

• Transizione da esecuzione ad attesa: *sospensione*
- avviene per esplicita azione del processo in esecuzione;
- generalmente, il processo che si sospende si pone in attesa di un evento *E.*

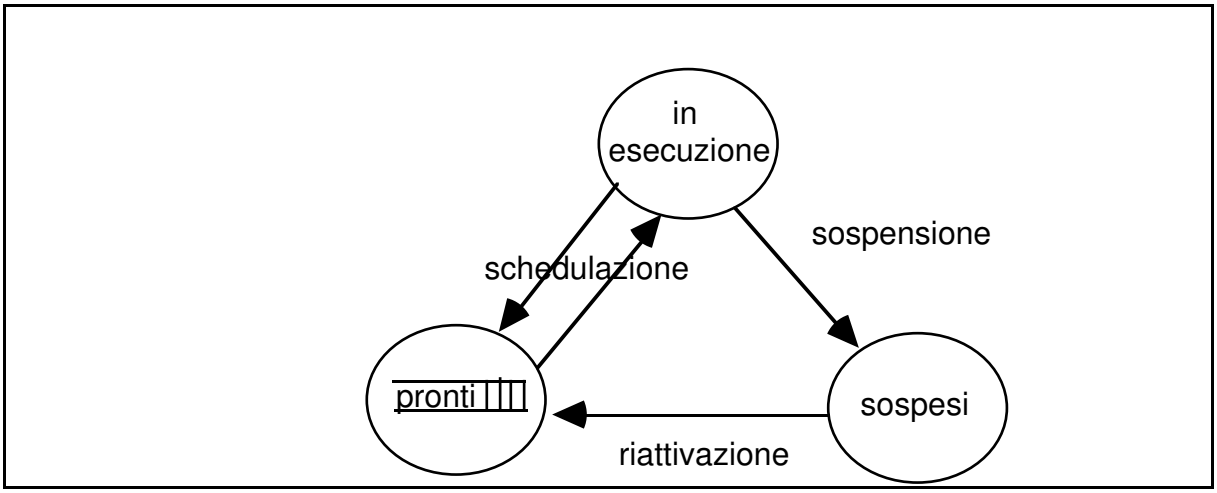
Per esempio,un processo Pi si sospende in attesa di:
- interruzione da parte di un dispositivo,
- sincronizzazione con un altro processo,
- ricezione di un messaggio

Il processo in attesa non è in competizione con i processi pronti; la riattivazione resta affidata alla cooperazione degli altri processi

• Transizione da stato di attesa a stato di pronto: *riattivazione:* avviene per esplicita azione di un processo P_j in esecuzione (eventualmente processo esterno).

- Schema tipico: P_i sospeso in attesa di evento *E,* viene riattivato da P_j che segnala *E.*

Stati e transizioni di stato



I processi non hanno alcun controllo sulle seguenti transizioni, che sono entrambe causate dalla funzione di gestione del processore (scheduler);

• Transizione da pronto a esecuzione: *assegnazione del processore*

• Transizione da esecuzione a pronto:
prerilascio (o revoca) del processore

Scheduler:
- controlla le velocità di avanzamento dei processi pronti
- utilizza il processore per fare avanzare i processi pronti (politica)

Ipotesi:
Uniprocessore: un processore ripartito in tempo tra n processi
Multiprocessore: # processi > # processori.

Dal punto di vista dei processi, nessuna differenza tra i due casi, salvo la diversa velocità di avanzamento.
Se fosse # processi= # processori, un processo potrebbe essere sempre attivo e le transizioni di stato non avrebbero motivo di esistere.

Gestione del processore (schedulazione)

L'avanzamento dei processi è determinato dalla disponibilità del processore e dalla politica con la quale l'uso del processore è ripartito tra i processi

La gestione del processore consiste nell'assegnare il processore ai processi pronti secondo una opportuna politica (schedulazione), avendo per obiettivo la massima utilizzazione del processore: quando il processo in esecuzione si sospende e passa in stato di attesa, il processore viene assegnato ad un altro processo pronto

Un altro obiettivo possibile consiste nel fare avanzare i processi con velocità dipendenti dalla loro *priorità*

Un ulteriore obiettivo può essere quello dell'equa ripartizione del processore tra i processi, evitando che alcuni di essi vengano ritardati indefinitamente a causa della precedenza accordata agli altri (fenomeno dell'*attesa indefinita).*

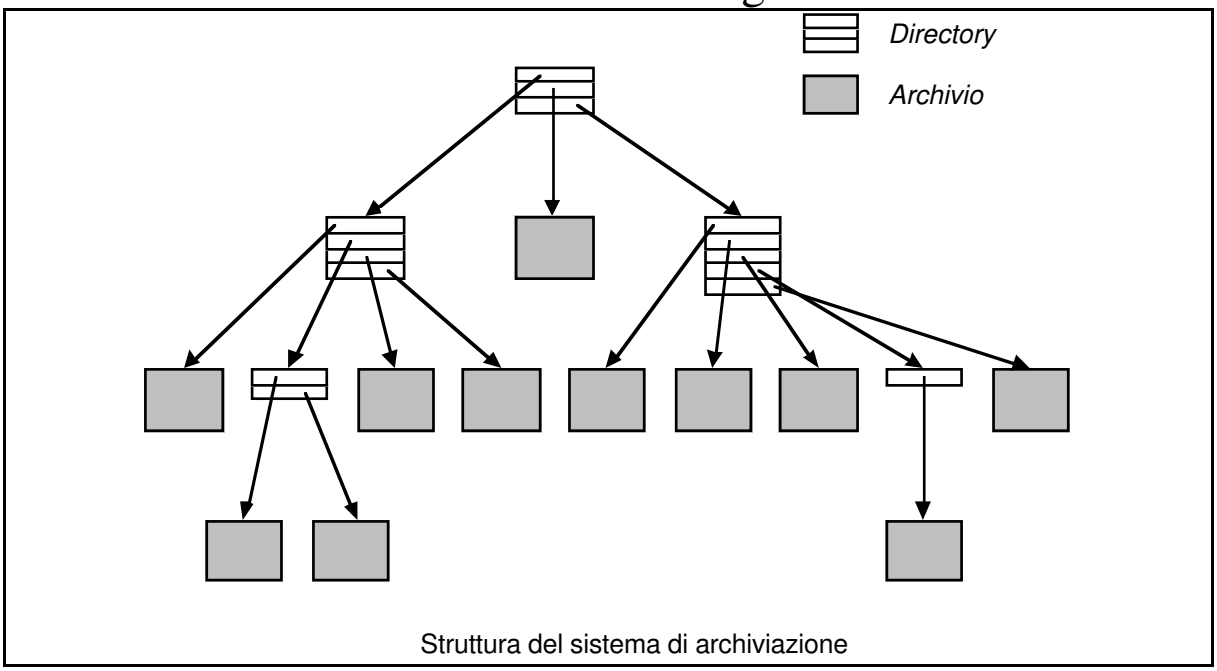
In genere il software di gestione del processore include anche la gestione delle interruzioni e dello scambio di messaggi tra processi

Gestione degli archivi

L'informazione contenuta negli archivi è memorizzata in dispositivi di memoria secondaria. Gli archivi possono essere distribuiti tra più dispositivi, anche di tipo diverso

Da un punto di vista logico un archivio è un segmento di informazione e ha una struttura logica contigua, ma la sua struttura fisica comprende tipicamente più blocchi, non necessariamente contigui.

Ad ogni file è associata una struttura *(descrittore)* che lo identifica e ne definisce la struttura logica e la struttura fisica



Struttura del sistema di archiviazione

Mediante il descrittore di archivio le operazioni definite sulla struttura logica possono essere tradotte in operazioni sulla struttura fisica, eseguibili dalla funzione di gestione dei dispositivi

Nei sistemi multi-utente, per ogni archivio sono definiti un *proprietario* e alcuni *diritti di accesso*, ad esempio per la lettura, la scrittura e l'esecuzione. Gli archivi possono essere condivisi da più utenti; utenti o gruppi di utenti diversi hanno generalmente diritti di accesso diversi.

Protezione

I sistemi operativi debbono proteggere le risorse condivise nei confronti di soggetti non autorizzati e delle operazioni che possono causare la perdita di informazione

La protezione consiste nel definire le limitazioni agli accessi e nell'imporre il rispetto di queste limitazioni mediante opportuni meccanismi.

I soggetti che sottostanno al sistema di protezione sono i **processi**

Nei sistemi di archiviazione, le limitazioni sono generalmente definite sotto forma di *diritti di accesso* associati agli archivi. I diritti di accesso per le diverse operazioni (tipicamente lettura, scrittura o esecuzione) possono essere attribuiti a singoli processi (tipicamente il proprietario dell'archivio) o a gruppi di processi (tipicamente un gruppo comprendente il proprietario) o alla totalità dei processi

La protezione dei **dispositivi** consiste nel riservare ai processi o alle procedure del s.o. la possibilità di eseguire le funzioni che agiscono sui dispositivi; la realizzazione è basata sulla distinzione tra stato utente o supervisore del processore e sui meccanismi che consentono di eseguire le *istruzioni privilegiate* solo quando il processore si trova in stato supervisore

La protezione della **memoria principale** consiste nel consentire a ogni processo di eseguire solo accessi interni al proprio spazio fisico

La protezione del **processore** ha l'obiettivo di impedire che un processo possa rimanere indefinitamente in stato di esecuzione

Gestione dei dispositivi

I dispositivi sono indipendenti, quindi asincroni rispetto al processore e interagiscono con esso attraverso il sistema di interruzione ed eventualmente con quello di *DMA*

I dispositivi periferici sono di gran lunga la componente meno affidabile di un calcolatore. In generale gli errori vengono riconosciuti attraverso la codifica ridondante dei dati e spetta al sistema operativo limitare le conseguenze. Una tipica strategia consiste nel ripetere l'operazione.

Nella gestione dei dispositivi si possono distinguere almeno tre distinte funzioni del sistema operativo
- il trattamento delle interruzioni
- il controllo dei dispositivi fisici
- la creazione di una interfaccia generale

Un dispositivo *D* è un processo esterno che interagisce con uno o più processi ordinari attraverso operazioni del tipo *signal(DBusy), wait(DBusy), signal(DReady), wait(DReady),* dove l'evento *DReady* è associato a un segnale di interruzione

A ogni dispositivo è associato un *processo di controllo (server)* che agisce per conto di tutti i processi che utilizzano il dispositivo

Gestione della memoria centrale

Bootstrap

Il caricamento in memoria centrale (MS) del sistema operativo *(bootstrap)* è basato su un piccolo programma *(caricatore primario)* residente in una porzione non volatile e a sola lettura della memoria principale, che il processore esegue automaticamente all'accensione.
Il caricatore primario copia in *MS* un programma contenuto un'area prestabilita di un dispositivo anch'esso prestabilito (ad esempio il primo blocco di un certo disco) ed esegue questo programma, che a sua volta copia in *MS* il sistema operativo
Dopo il caricamento primario, la porzione della memoria non occupata dal sistema operativo *(memoria disponibile)* è disponibile per i processi ed è gestita dalla funzione di gestione della memoria.

Gestione statica o dinamica

Si parla di gestione statica o di gestione dinamica della memoria a seconda che il caricamento dei programmi sia statico o dinamico
La struttura dati principale, sebbene sia diversa a seconda del modello di caricamento e della sua realizzazione, verrà sempre indicata con il nome di *descrittore della memoria*

La funzione di gestione della memoria decide sui caricamenti e sugli scaricamenti in base alle informazioni contenute nel descrittore della memoria e in altre strutture dati.
L'esecuzione dei comandi di caricamento o scaricamento generati dalla funzione di gestione della memoria compete alle funzioni di gestione dei dispositivi.