



Sistemi Operativi

Fabio Cantaro

CLASSIFICAZIONE DEL SOFTWARE

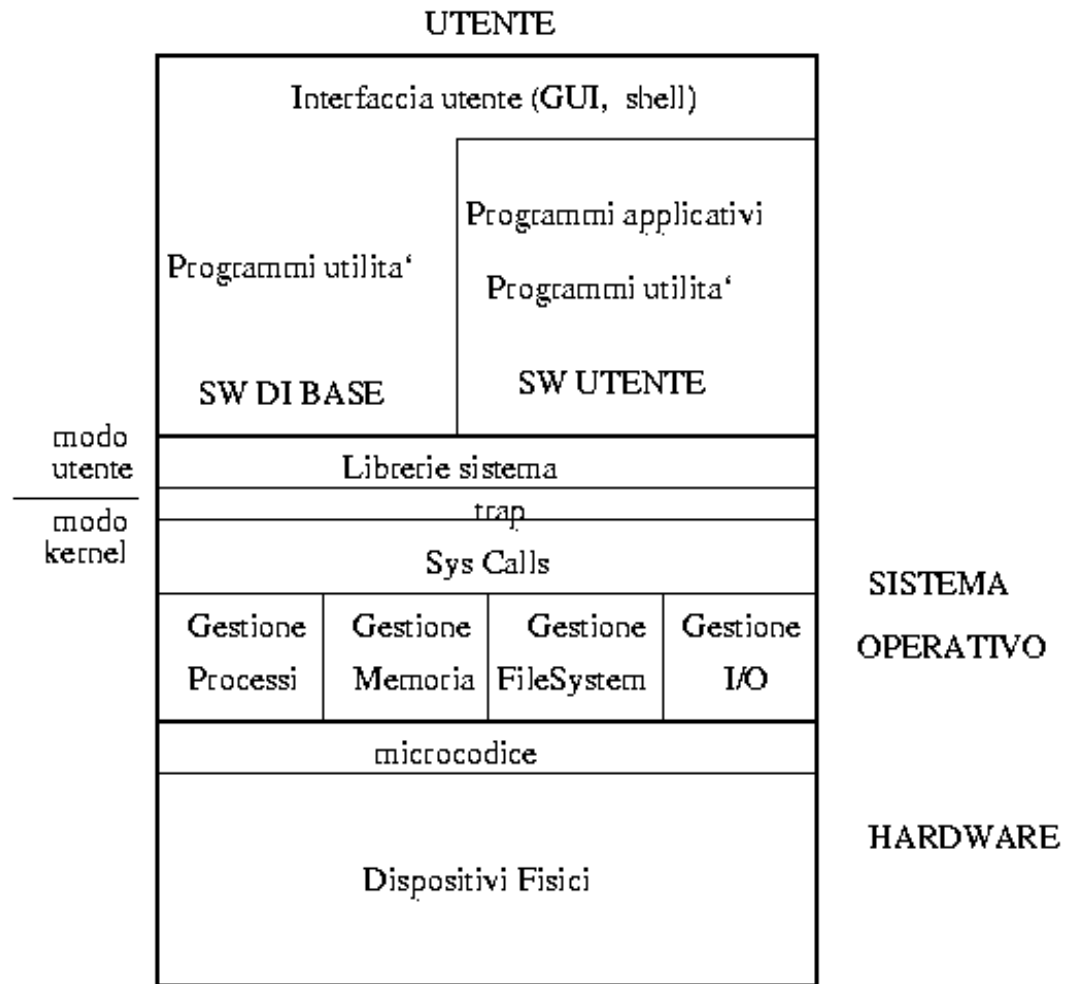
TIPI DI SOFTWARE

Per software si intende l'insieme delle istruzioni che guidano l'hardware nella esecuzione dei suoi compiti.

In prima approssimazione possiamo classificare il software nelle seguenti categorie:

- **Software di base** è l'insieme di programmi che permettono il governo di un elaboratore e che solitamente vengono forniti assieme all'elaboratore. Del software di base fanno parte:
 - **il sistema operativo**, ovvero quello strato di software, tra l'hardware e l'utente, che permette il funzionamento di un computer, ne gestisce le risorse e ne nasconde la complessità al software applicativo.
 - **programmi di utilità** necessari per agevolare l'utente nell'interazione con il computer e l'amministratore nella configurazione e nella gestione del sistema operativo. Le distribuzioni di Linux (Ubuntu, Redhat, Caldera, Suse, ecc) e Windows nelle sue varie forme (Windows 2000, XP, ME ecc) coprono la quasi totalità delle installazioni nei sistemi desktop e server.
- **Software applicativo** è quel software che viene utilizzato per svolgere del lavoro utile su un computer (produzione e gestione di documenti, database, strumenti per il calcolo, multimedia, ecc.)

CLASSIFICAZIONE DEL SOFTWARE



LICENZE D'USO

Il software utilizzato sui calcolatori può essere classificato anche in base al tipo di licenza d'uso:

- **Software commerciale:** software sviluppato da un'azienda allo scopo di guadagnare dal suo uso. Ad esempio gran parte software Microsoft (Windows , MS Office, VisualStudio, ecc) e' commerciale.
- **Freeware e Shareware:** Software distribuito senza codice sorgente e che può essere ridistribuito (tipicamente via Internet).
 - Il Freeware non prevede costi nell'utilizzo. La licenza Freeware viene molto utilizzata dalle aziende come manovra per vendere software commerciale (versione "Professional") propagandato mediante la distribuzione Freeware di una versione con caratteristiche limitate (spesso denominata "Light"). Ad esempio Acrobat e' software commerciale, mentre "Acrobat Reader" (permette solo la lettura di documenti PDF) e' Freeware
 - Lo Shareware impone un costo per il suo uso dopo un periodo di prova..
- **Open Source:** sotto questo nome si intendono diversi tipi di licenze che hanno però il comune obiettivo di diffondere liberamente lo scambio di idee e informazioni e vengono quindi utilizzate prevalentemente negli ambiti scientifici per software senza fini di lucro. Le regole principali sono: il software deve essere libero, deve includere il codice sorgente (Open Source) e deve essere modificabile e ridistribuibile.



Sistema Operativo

Un sistema di elaborazione è composto da vari dispositivi.
=> Gestirli in maniera ottimale è difficile: servono i sistemi operativi.

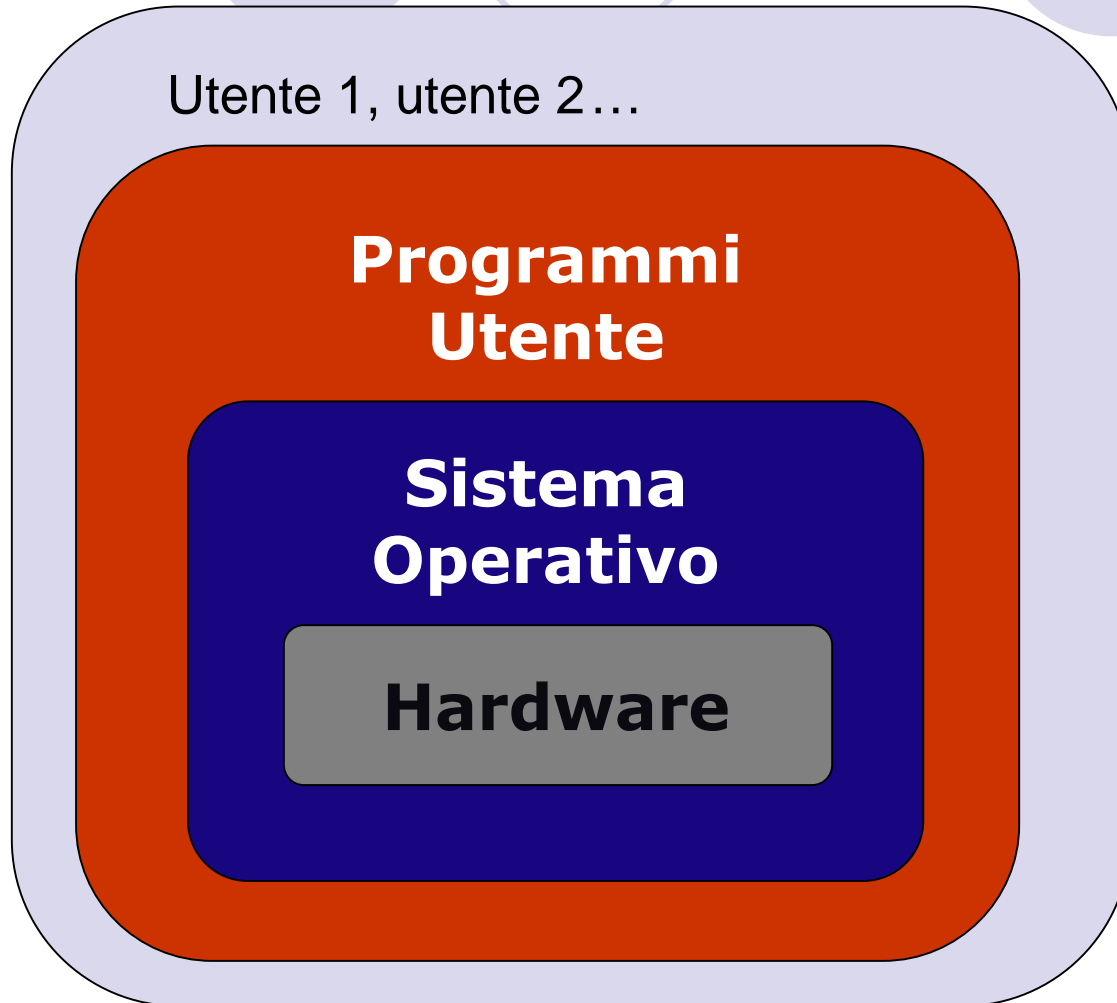
- il SO è un Insieme di programmi base che costituiscono l'interfaccia logica tra l'utente e l'hardware.
- Il SO nasconde l'hardware non solo all'utente ma anche ai programmi.
- Permette una gestione ottimizzata del hard disk, della CPU, della memoria principale, e delle periferiche quali floppy disk, stampante, schede di espansione, etc.

Sistema Operativo

Un Sistema Operativo deve:

- Permettere all'utente di usare in modo semplice e immediato tutte le risorse che il sistema di elaborazione mette a disposizione;
- Ottimizzare l'uso delle risorse a disposizione, attuando opportune politiche di gestione,
- Fornire un'immagine astratta di tutte le risorse disponibili

Sistema Operativo



- L'utente non interagisce direttamente con l'hardware ma con un sistema (MACCHINA ESTESA) composto da hw e SO che gestisce le risorse reali
- => mentre i moduli del SO sono eseguiti dalla Macchina reali (hw), i programmi utente su una macchina estesa

Funzioni del Sistema Operativo

Il SO deve:

- Gestire in maniera efficiente (politiche di gestione) le risorse del sistema:
 - l'unità centrale
 - la distribuzione della memoria fra più programmi
 - le unità di input/output
 - le operazioni di lettura e scrittura sulle memorie di massa

- rendere agevole il dialogo con la macchina

- fornire un'immagine astratta di tutte le risorse disponibili, indipendentemente da come esse funzionano

- rendere possibile il collegamento in Rete

Sistema Operativo



- La funzionalità del sistema operativo è quella di presentare all'utente una macchina estesa o virtuale che sia più facile da programmare che non l'hardware sottostante.
- Immaginate cosa succederebbe se tre programmi che girano su calcolatore tentassero di usare contemporaneamente la stessa stampante. Il sistema operativo risolve bufferizzando tutti i dati su un disco.

Sistema Operativo



- Il SO dà l'impressione all'utente di poter comunicare direttamente con ogni aspetto del PC,
- in realtà è sempre lui che agisce da anello di congiunzione tra utilizzatore e macchina.
- Tutto ciò è evidente con i moderni sistemi operativi dotati di interfaccia grafica ([GUI](#)), gestibile con un mouse.
- In passato, quando ancora si usavano sistemi detti "a riga di comando", l'[interazione](#) dell'utente con il sistema operativo era più visibile e più complessa;
- dal [prompt](#) era necessario impartire precisi comandi dotati di parametri propri direttamente al sistema, che rispondeva semplicemente con un messaggio.
- È questo il caso del Microsoft [DOS](#)



Storia dei sistemi operativi

Parallela alla storia dei sistemi di elaborazione:
prima generazione, 1945-55

- computer costruiti con relay e valvole: poco affidabili
- il progettista, programmatore e utente sono la stessa persona
- programmazione in linguaggio macchina, tramite pannelli con contatti fisici prima e schede perforate dopo
- quasi nessun sistema operativo: solo driver dispositivi e bootstrap



Storia dei sistemi operativi

seconda generazione, 1955 -65

- nasce il transistor: affidabili, possono essere venduti a terzi: mainframes
- progettista, programmatore ed operatore sono figure diverse
- programmazione in assembler o fortran tramite schede perforate
- sistemi operativi batch: l'operatore raccoglie i job presentati dagli utenti e li esegue in rapida successione



Storia dei sistemi operativi

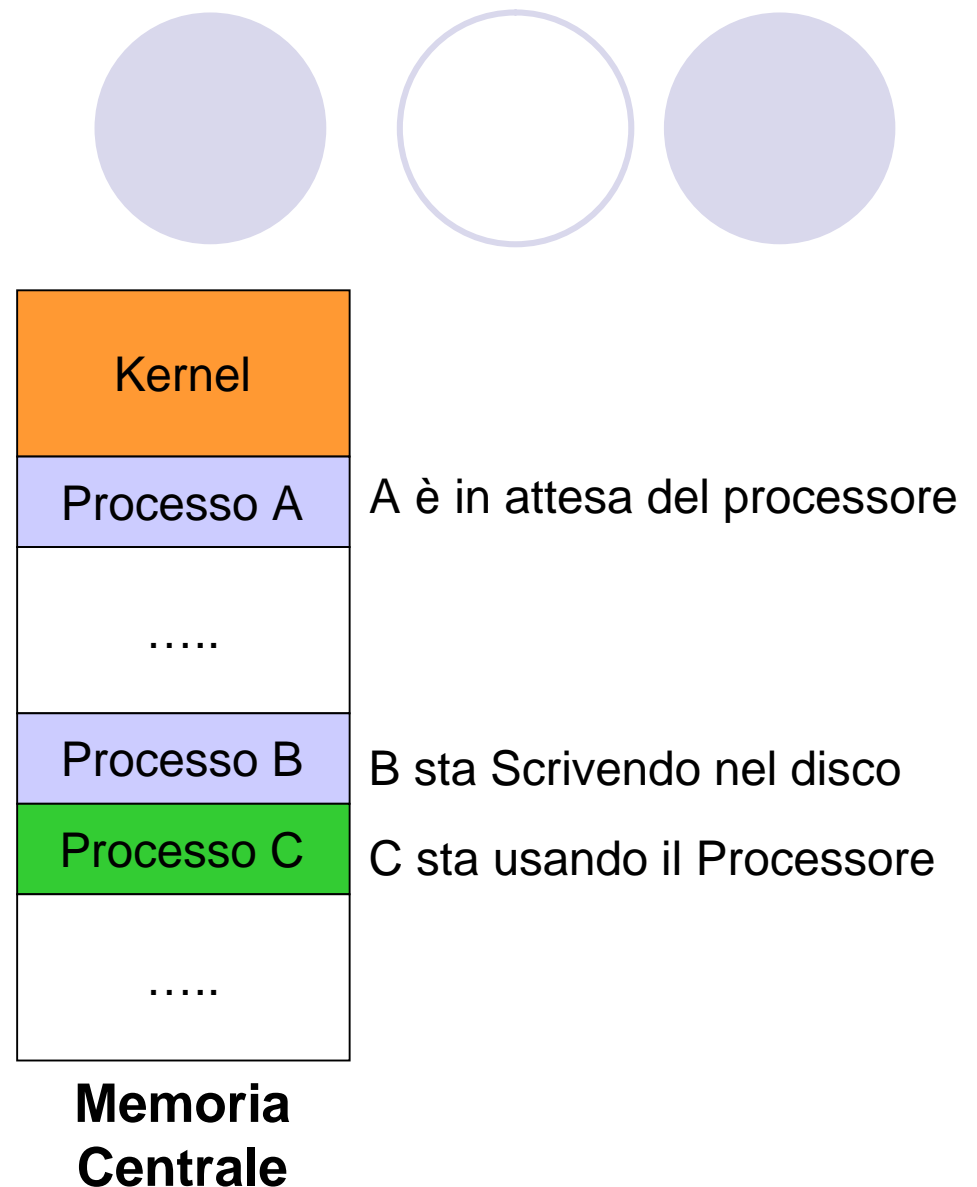
terza generazione, 1965-80

- circuiti integrati: diffusione negli ambienti commerciali, minicomputer.
- **spooling** (prima) e multiprogrammazione (poi) per ridurre gli sprechi di tempo nelle operazioni di I/O
- **Multitasking**: permette l'esecuzione contemporanea di più programmi (residenti contemporaneamente in memoria in Aree Distinte).
 - Problemi di protezione: proteggere i programmi degli utenti da intrusioni
 - Problemi di condivisione delle risorse tra i vari programmi in memoria

MULTITASKING

Quando un task (esempio task B) esegue una operazione di I/O

=> la CPU interrompe l'esecuzione e passa al task successivo (esempio task C).

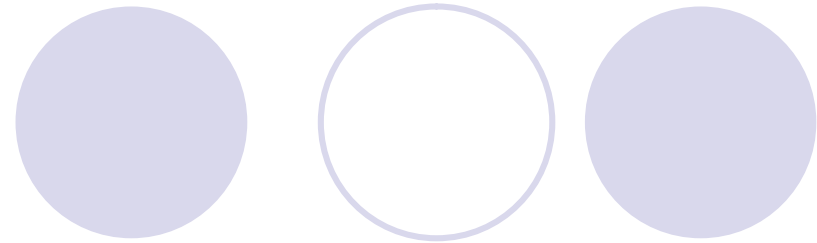


Storia dei sistemi operativi

terza generazione, 1965-80

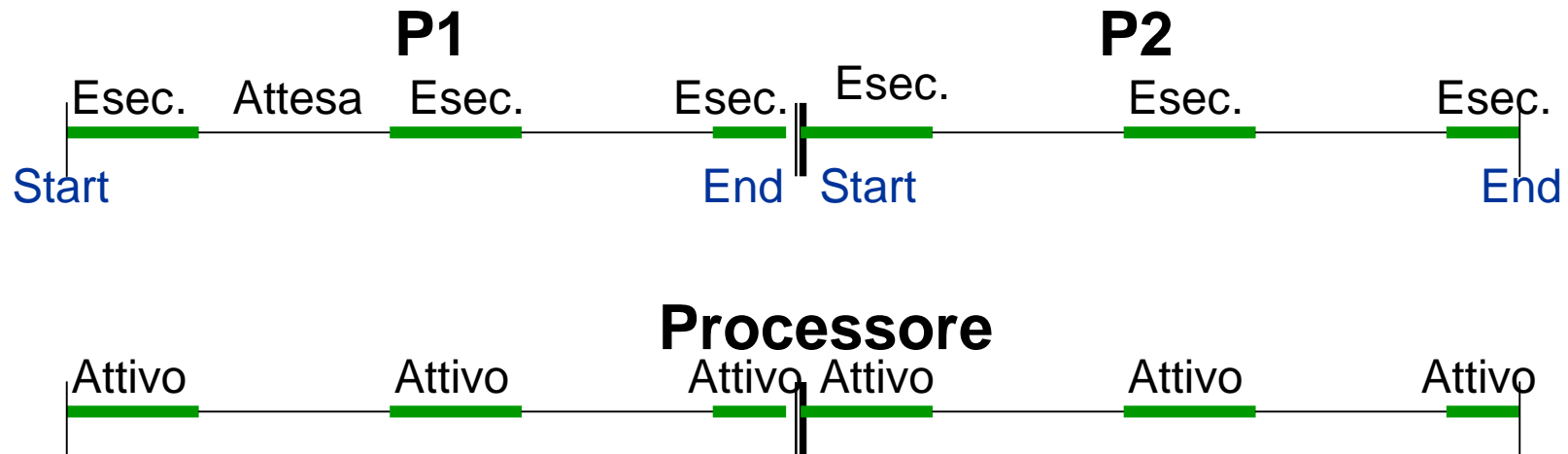
- **timesharing** per avere più utenti contemporaneamente. L'idea era quella di ridurre i tempi morti del processore.
 - Nasce il concetto di Quanto di Tempo :
 - anche se un task se non incontra operazioni di Input/Output entro un certo tempo viene egualmente sospeso dall'esecuzione.
 - => ogni utente si illude di avere tutto il sistema a propria disposizione.
 - Però le operazioni di CONTEXT SWICHT appesantiscono il sistema

Time Sharing(1)

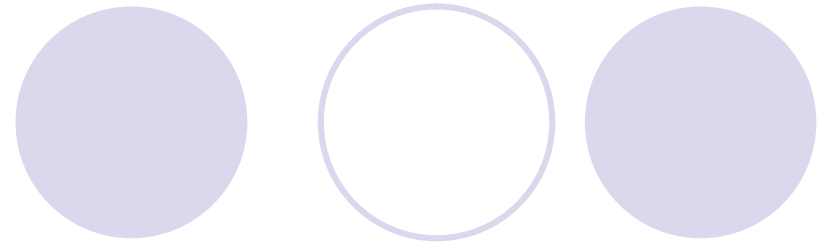


INEFFICIENZA DELL'ESECUZIONE SEQUENZIALE:

prima del Time Sharing ogni processo occupava il processore anche nei periodi di inattività

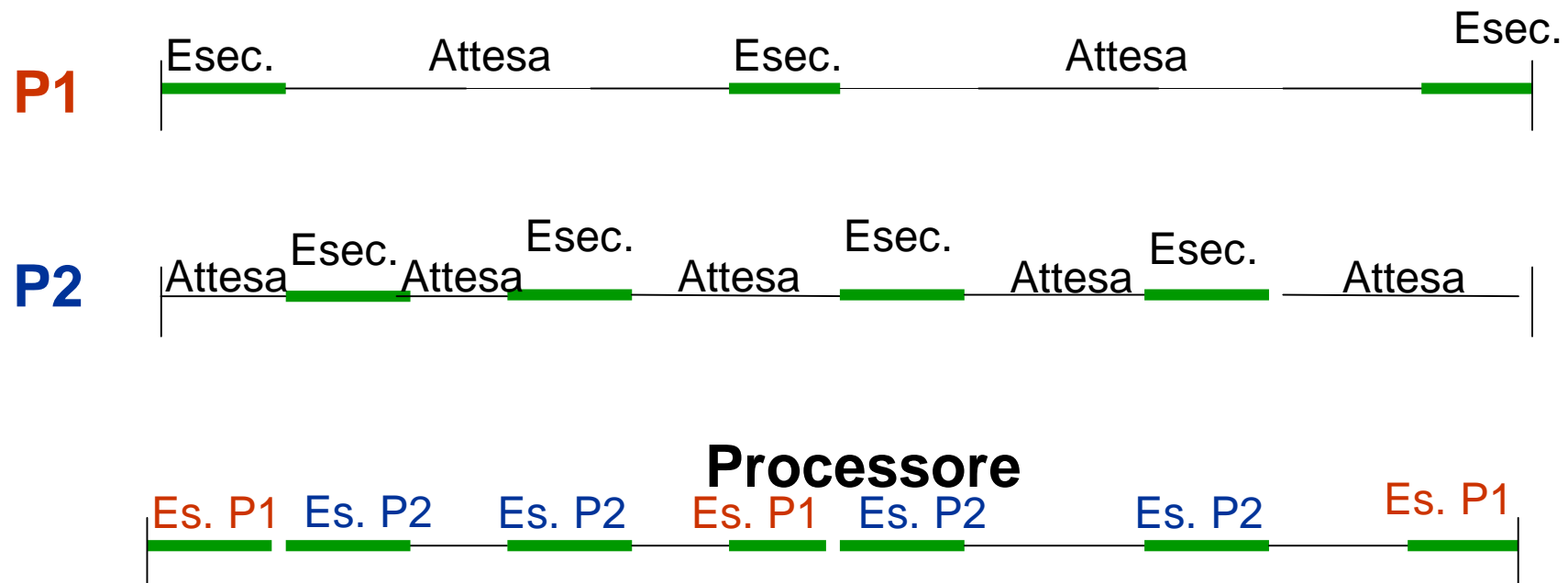


Time Sharing(2)



EFFICIENZA DELL'ESECUZIONE CONTEMPORANEA DI P1 e P2:

col Time Sharing il processore veniva assegnato a più processi.





Storia dei sistemi operativi

quarta generazione, 1980-

- circuiti integrati LSI (large scale integration): microcomputer
- interfacce utente grafiche (**GUI**)
- gestione di reti di elaboratori (**network operating systems**)



I sistemi operativi ai giorni nostri

- sistemi operativi per mainframe: IBM

OS/390

- gestione di più processori fortemente accoppiati
- come personal computer ma con immense capacità in termini di I/O
- (migliaia di dischi);
- elaborazioni batch, timesharing



I sistemi operativi ai giorni nostri

- sistemi operativi per server: **Unix/Linux,**
Windows NT/2000/XP
 - girano su personal computer ma anche su mainframe
 - gestione di più processori fortemente accoppiati
 - compiti primari: file server, print server, web server
 - elaborazioni batch, timesharing

I sistemi operativi ai giorni nostri

- sistemi operativi per personal computer:
Linux, Windows 98/XP
 - interfaccia utente
- sistemi operativi real-time:
 - requisiti temporali che devono essere rispettati (conflitto con time-sharing)

I sistemi operativi ai giorni nostri

- **sistemi operativi distribuiti**

- distribuzione della computazione su più macchine:

- Strettamente accoppiati: i processori condividono il clock di sistema e la memoria
- Debolmente accoppiati: ogni processore dispone di propria memoria locale (network operating systems => reti di elaboratori) visione unitaria del sistema di calcolo

- tolleranza ai guasti, condivisione risorse, aumento della velocità

- **sistemi operativi per palmtop** (PalmOS, Windows CE)

- **sistemi operativi per smart-card**

Tipi di Sistema Operativo

- mono-utente: se non è in grado di distinguere gli utenti che utilizzano il sistema. Un solo utente alla volta.
- mono-task: se ammette in esecuzione un solo programma per volta.
- I sistemi operativi dei primi calcolatori (anni 50) erano mono-utente e mono-task. Ma anche MsDos.
- multi-task: un solo utente che può però eseguire contemporaneamente più programmi. (Es. Windows98/ME e MacOS)
- Multiutente (multiuser): SO che permettono a più utenti di usare lo stesso computer:
 - Al computer sono collegati più terminali (tastiera e video)
 - Tutti gli utenti condividono le stesse risorse (cpu, memoria, ...)
 - Gestione trasparente delle risorse: ogni utente deve avere l'impressione di essere il solo a utilizzare il computer.
 - SO Multiuser più comuni: Windows NT/2000/XP, Unix (con diversi "dialetti": Aix, Solaris, HPux, True64Unix, Linux, ecc).

Concetti di Base

- **Processo (Task)**: si intende un programma in esecuzione
 - durante l'esecuzione il calcolatore passerà da uno stato all'altro => ogni stato sarà caratterizzato da:
 - contenuto (dati o indirizzi) di un certo numero di Registri Generali
 - Program Counter
 - Registro di stato, tra le altre informazioni ci dirà:
 - Se sta eseguendo un'istruzione di sistema (Stato SuperVisore)
 - O se sta eseguendo un'istruzione Utente (Stato Utente)
- un **programma** è invece una sequenza statica di istruzioni
- **Risorsa**: qualsiasi elemento hw/sw (Cpu, memoria, ecc...) usato da un processo e che ne condiziona l'avanzamento
- **FILE**: Archivio in memoria di massa

Sistema Operativo: Gestore di Risorse

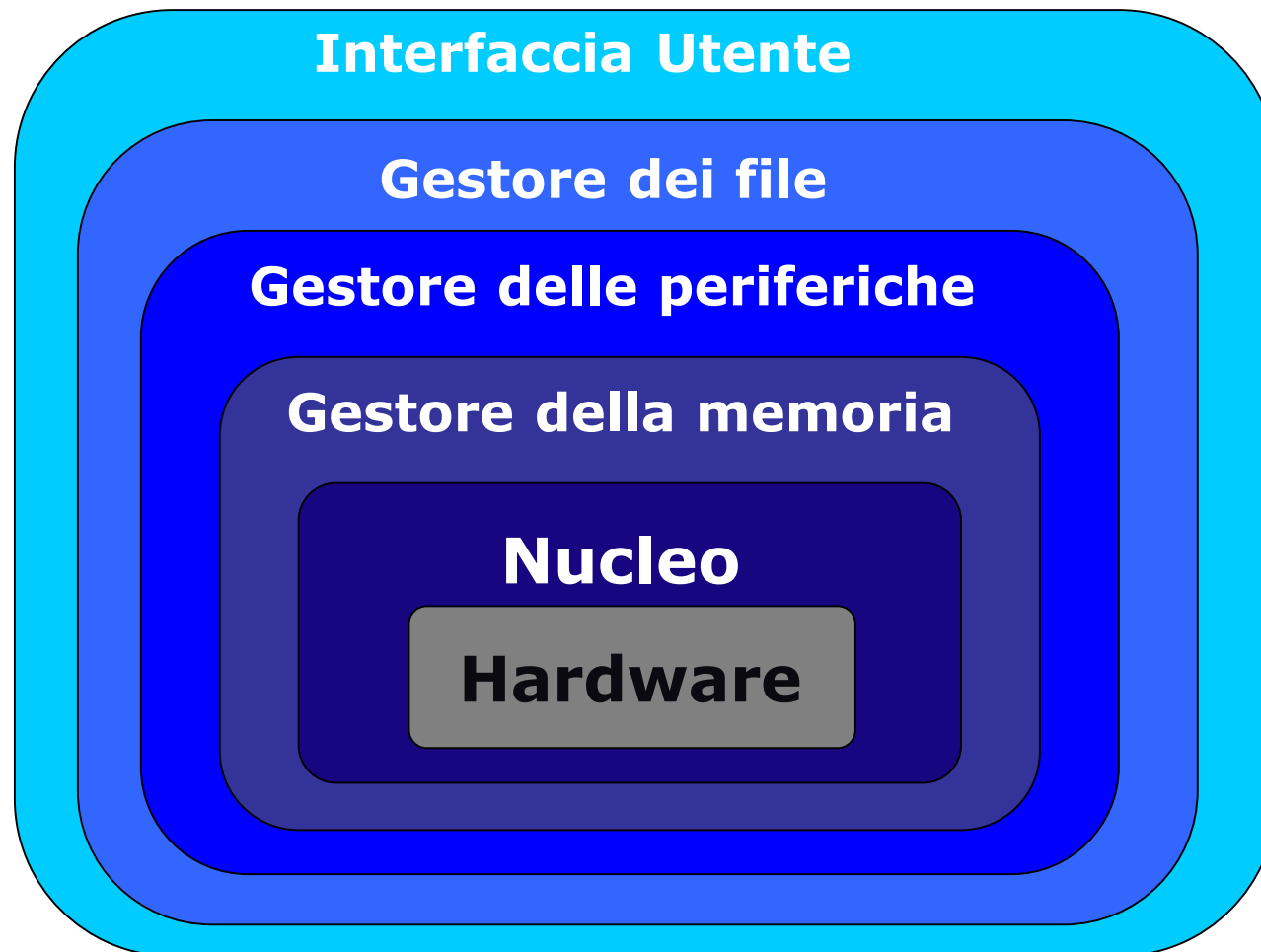
- Le operazioni logiche che il S.O. deve compiere per gestire correttamente le risorse del sistema sono:
 - Tenere aggiornate le informazioni sullo stato di ciascuna risorsa;
 - Interpretare le richieste dei singoli processi;
 - Assegnare la singola risorsa a un determinato processo;
 - Riprendere il controllo della risorsa.

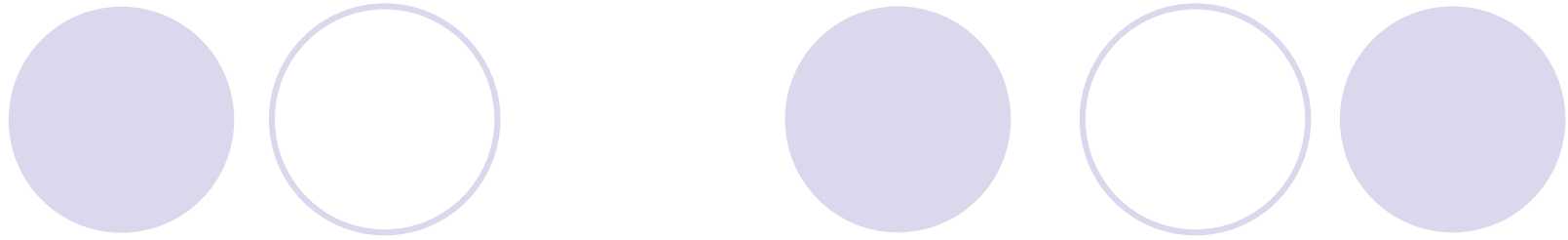
Sistema Operativo: Gestore di Risorse

- In un SO Multitasking più processi possono richiedere stesse Risorse => il SO deve risolvere eventuali conflitti (**Competizione**)
- Il SO decide “cosa” (ogni risorsa) assegnare a “chi” in base a politiche di Schedulazione
- I moduli di SO dedicati alla gestione delle risorse sono:
 - **Gestore dei Processi**, che si occupa dell’assegnazione del processore centrale ai singoli processi;
 - **Gestore della memoria**, che gestisce la memoria centrale del sistema;
 - **Gestore delle periferiche**, che gestisce tutte le periferiche;
 - **Gestore delle informazioni**, che si occupa della memorizzazione e del recupero delle informazioni, archiviate nella memoria di massa.

Struttura del sistema operativo


- Una struttura a livelli
- Ogni livello "aggiunge" qualcosa





IL NUCLEO

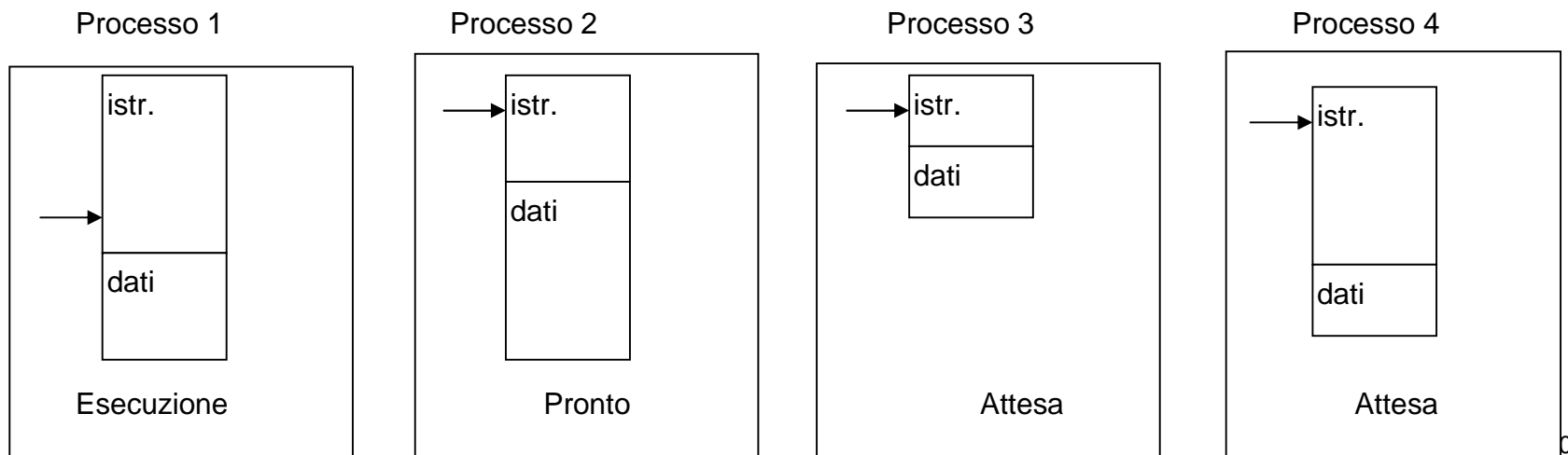
Il Nucleo (Kernel)



- Comprende i moduli del SO per la gestione del Processore
- È la parte più vicina all'hardware
- Le sue funzioni sono:
 - avvio e terminazione dei processi
 - assegnazione della CPU ai vari processi attraverso politiche di Scheduling.
 - sincronizzazione tra i processi

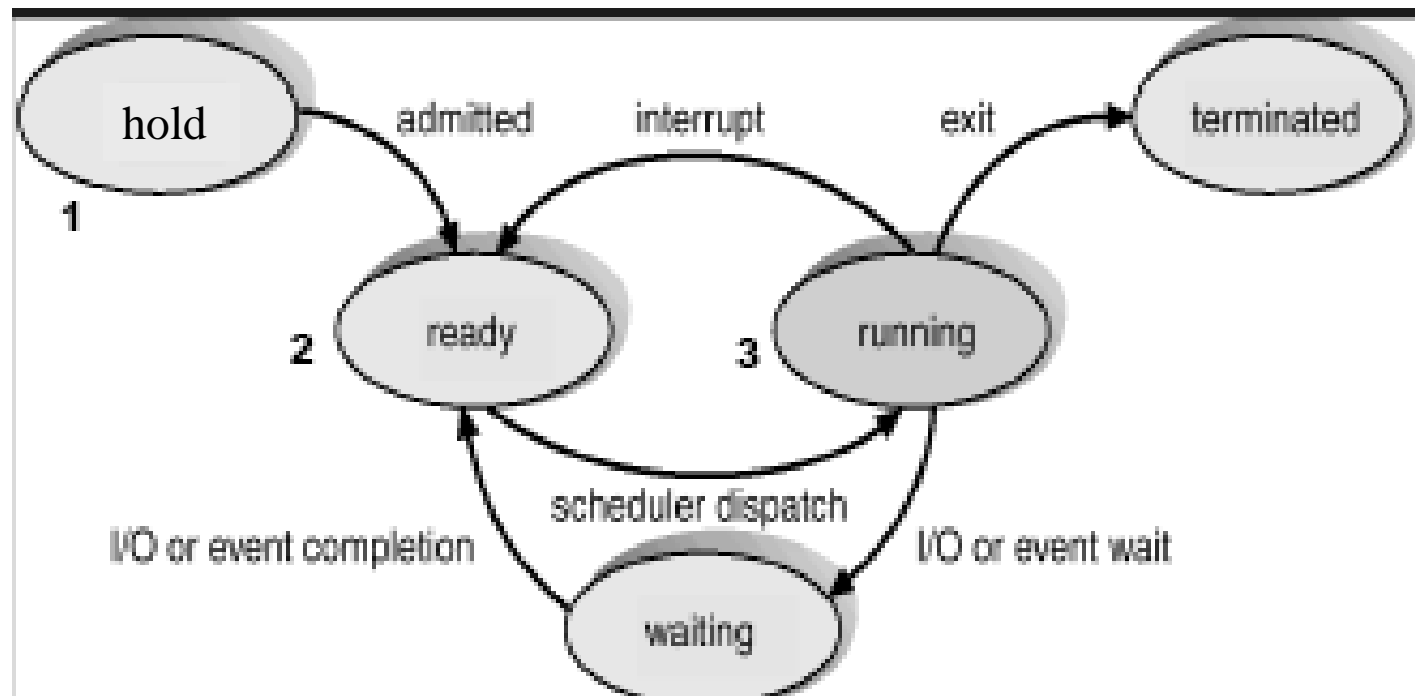
Il Nucleo: gestore dei processi

- Il passaggio tra un processo e l'altro in esecuzione sulla CPU è gestito dal sistema operativo mediante il context switching (Scambio del contesto).
- Il contesto è una "fotografia" dello stato della CPU che viene memorizzato al momento dello scambio, per poi essere ripristinato quando il processo tornerà in esecuzione.
- L'informazione più importante del Contesto è il "Program Counter" ovvero l'indirizzo della prossima istruzione che il processo dovrà eseguire.



Il Nucleo: gestore dei processi

- Il sistema operativo dovrà gestire anche lo stato del processo relativamente alla competizione per l'accesso al processore => Per questo esistono 5 stati possibili in cui può trovarsi un processo:



Gestore dei processi



In generale gli stadi di avanzamento di un processo possono essere sintetizzati come:

- Disponibilità (**hold**): il nome del programma è inserito nell'elenco di quelli in attesa di esecuzione.
- Pronto (**ready**): il processo ha a disposizione tutto ciò che gli serve (risorse) per essere seguito, ma deve aspettare il proprio turno (politiche di schedulazione)
- Esecuzione (**run**): il processo è in esecuzione sul processore
- Attesa (**wait**): il processo non può avanzare in quanto non ha a disposizione tutte le risorse necessarie
- Terminazione (**complete**): il processo è terminato e tutte le risorse ad esso assegnate vengono rilasciate

Gestore dei processi: Politiche di Scheduling

Una buona politica di scheduling deve:

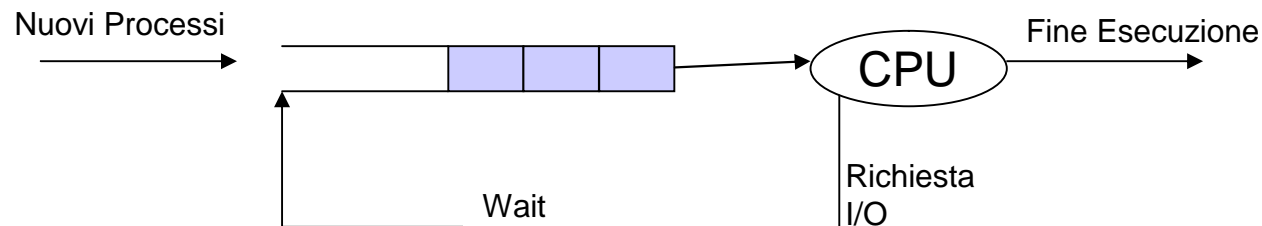
- Massimizzare l'uso della CPU: processore attivo il più possibile
- Massimizzare il Throughput: cioè il numero di processi serviti nell'unità di tempo
- Minimizzare il TurnAround: tempo di attività del processo dalla creazione alla terminazione
- Minimizzare il Tempo di Attesa che un processo passa nella coda dei pronti
- Minimizzare l'OverHead di sistema: costo da pagare per il Context Switch

Politiche di Scheduling

Due categorie di politiche:

1. **Non Preemptive**: un processo in esecuzione non è interrompibile. Esempi di politiche NP:

- **First In First Out**: Processore assegnato al 1° nella coda dei pronti



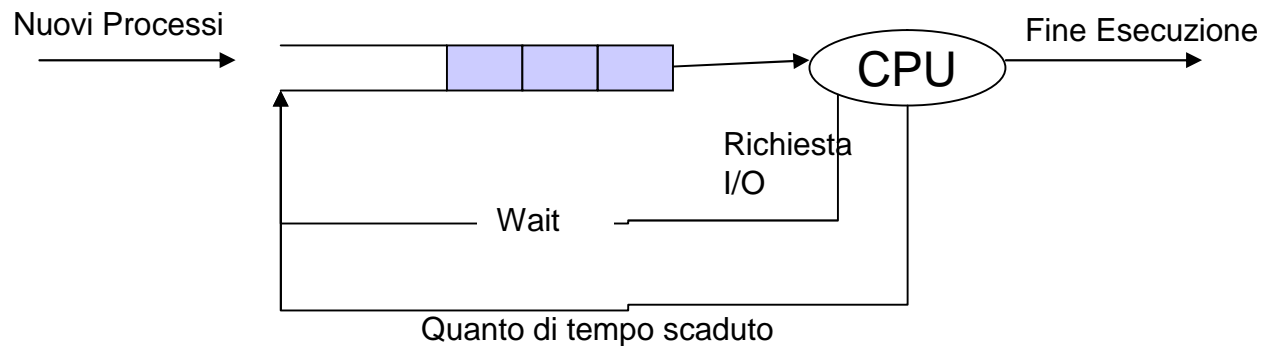
- **Shortest Job First**: Processore assegnato al processo più corto tra quelli pronti

Politiche di Scheduling

2. **Preemptive**: la CPU può essere tolta ad un processo in esecuzione.

Esempio di politiche Preemptive:

- Round Robin: processore assegnato a turno (quanto di tempo) ai processi pronti.

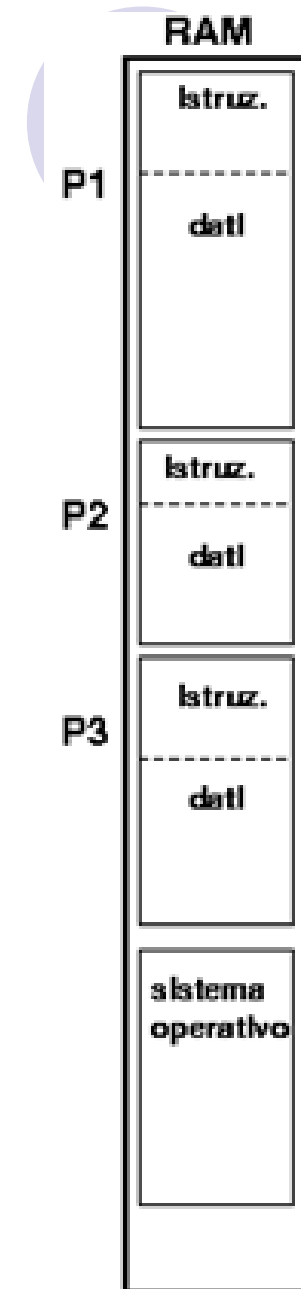




IL GESTORE DELLA MEMORIA

Memoria

- Ogni processo per essere eseguito dev'essere residente in memoria (spazio degli indirizzi)
- Spazio logico: l'immagine di un processo è logicamente composta da sezioni (codice e dati) che possono essere viste dal processo come contigue (anche se fisicamente disgiunte in memoria fisica)
- Una parte della memoria è riservata al Sistema Operativo, la rimanente è quella disponibile per i processi.





Gestore della memoria

- Il G.M. è un modulo di SO che deve essere eseguito
- Il G.M. deve gestire la memoria in modo da:
 - Decidere a quale processo assegnare memoria, con che priorità e per quanto tempo
 - non limitare la possibilità di creare altri processi
 - Consentire l'avanzamento di tutti i processi
 - Tenere aggiornato lo stato di ogni locazione di memoria

Gestore della memoria

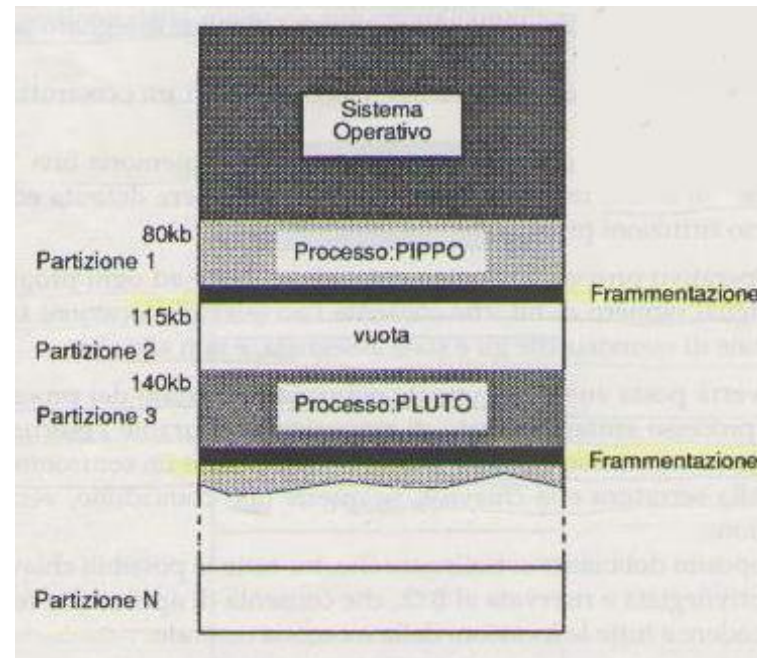
Esistono molte politiche di gestione della memoria:

- **Allocazione a singola partizione** : utilizzata per SO NON Multitasking. Problema: Molto spreco di memoria (non usata dal processo in memoria)
- **Allocazione a partizioni** (gestione semplice per SO Multitasking):
 - **Statica**: la memoria veniva divisa in Partizioni Fisse e poi i processi venivano allocati (Problema: Frammentazione Interna)
 - **Dinamica**: ogni volta che si carica un programma il SO crea una partizione ad hoc (“cucita addosso”). Problema: Frammentazione esterna. Ogni tanto Deframmentazione (Defrag in Windows)
- **Paginazione**
- **Segmentazione**

Gestore della memoria



Allocazione a Singola partizione

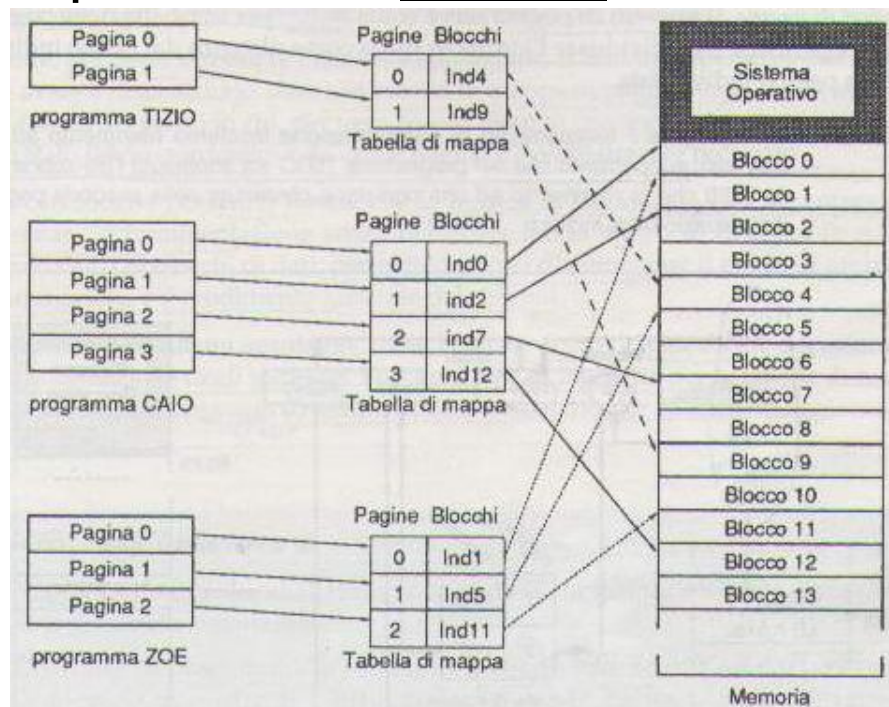


Allocazione a Partizione Multipla

Gestore della memoria

Paginazione

- Fin'ora allocazione contigua. La Paginazione permette una allocazione NON CONTIGUA, cio è l'immagine (spazio degli indirizzi) del processo viene spezzata in PAGINE
- Esiste una corrispondenza biunivoca tra le pagine del processo e Blocchi di memoria che li devono contenere

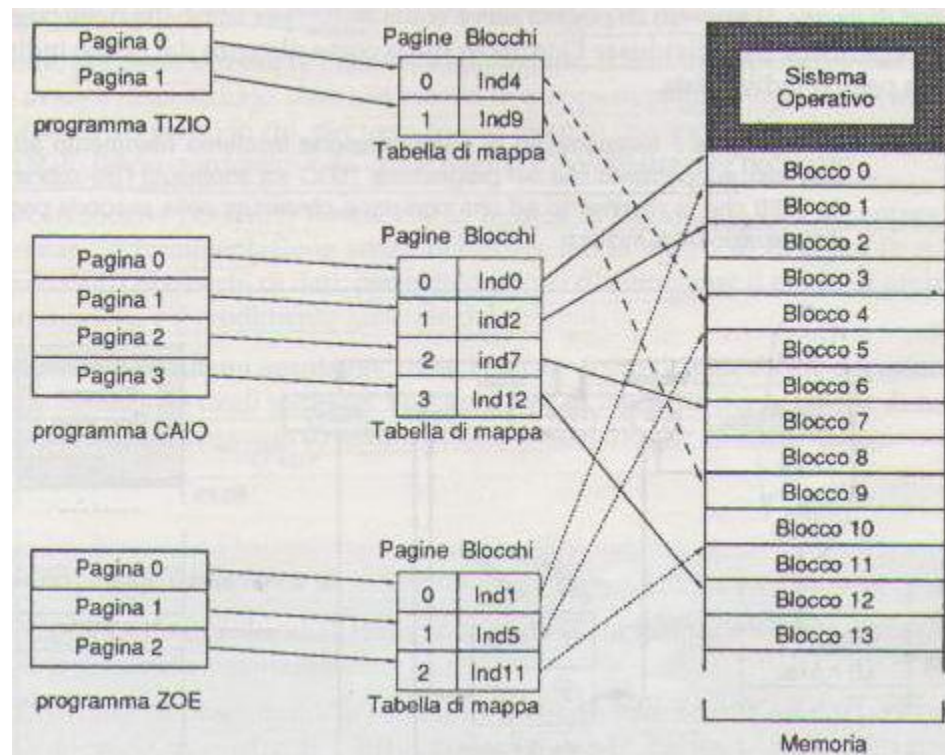


- L'utente vede il processo in pagine logicamente contigue ma in memoria fisica in pagine Non Contigue
- Dimensioni Tipiche pagina: 4k
- si utilizza una TABELLA DI MAPPA per trasformare l'indirizzo di Pagina in INDIRIZZO FISICO DI MEMORIA

Gestore della memoria

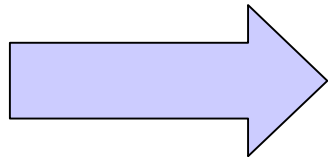
Esempio di Paginazione

- Il programma Caio lungo 14 K => viene suddiviso automaticamente in 4 pagine da 4 k (ultima non completamente occupata => piccola frammentazione interna)
- In memoria vengono reperiti 4 blocchi liberi: (0,2,7,12)



Memoria Virtuale

- La Memoria fisica può NON ESSERE abbastanza grande da contenere l'intero spazio degli indirizzi di un processo
- Alcune parti di un programma vengono usate raramente (particolari procedure nei programmi a menù oppure Routine di gestione degli errori)



Soluzione: Caricare in memoria solo una parte del programma, lasciando la rimanente in Memoria di massa

Memoria Virtuale



Vantaggi

- Un programma non è più vincolato dalla quantità di memoria fisica disponibile => si possono scrivere prog molto grandi
- Si può aumentare il Throughput di sistema => aumento del grado di multiprogrammazione, visto che si possono far avanzare più programmi

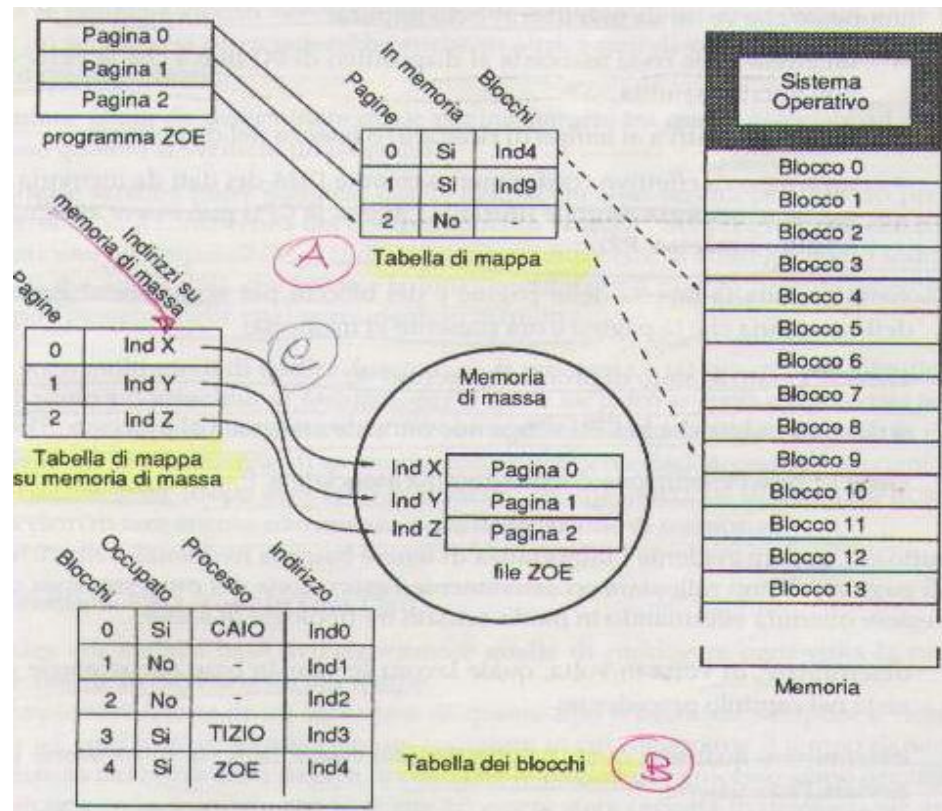
Organizzazione del gestore

- **Tabella di mappa di pagina** (per ogni processo): informazioni se ad una pagina logica è assegnato un blocco fisico di memoria oppure se è in Memoria di Massa
- **Tabella dei Blocchi** (unica per tutto il sistema): per ogni blocco dice se è occupato e da chi
- **Tabella di Mappa SU DISCO** (per ogni processo): ad ogni pagina è associato l'indirizzo fisico in memoria di massa

Memoria Virtuale

Per ogni processo viene creata A e C:

- **Tabella di mappa di pagina (A)** (per ogni processo): informazioni se ad una pagina logica è assegnato un blocco fisico di memoria oppure se è in Memoria di Massa
- **Tabella dei Blocchi (B)** (unica per tutto il sistema): per ogni blocco dice se è occupato e da chi
- **Tabella di Mappa SU DISCO (C)** (per ogni processo): ad ogni pagina è associato l'indirizzo fisico in memoria di massa



Memoria Virtuale



- Mediante la **Paginazione** in sistema operativo fornisce ad ogni processo una area di memoria “virtuale”.
Per macchine a 32 bit quest’area può essere di

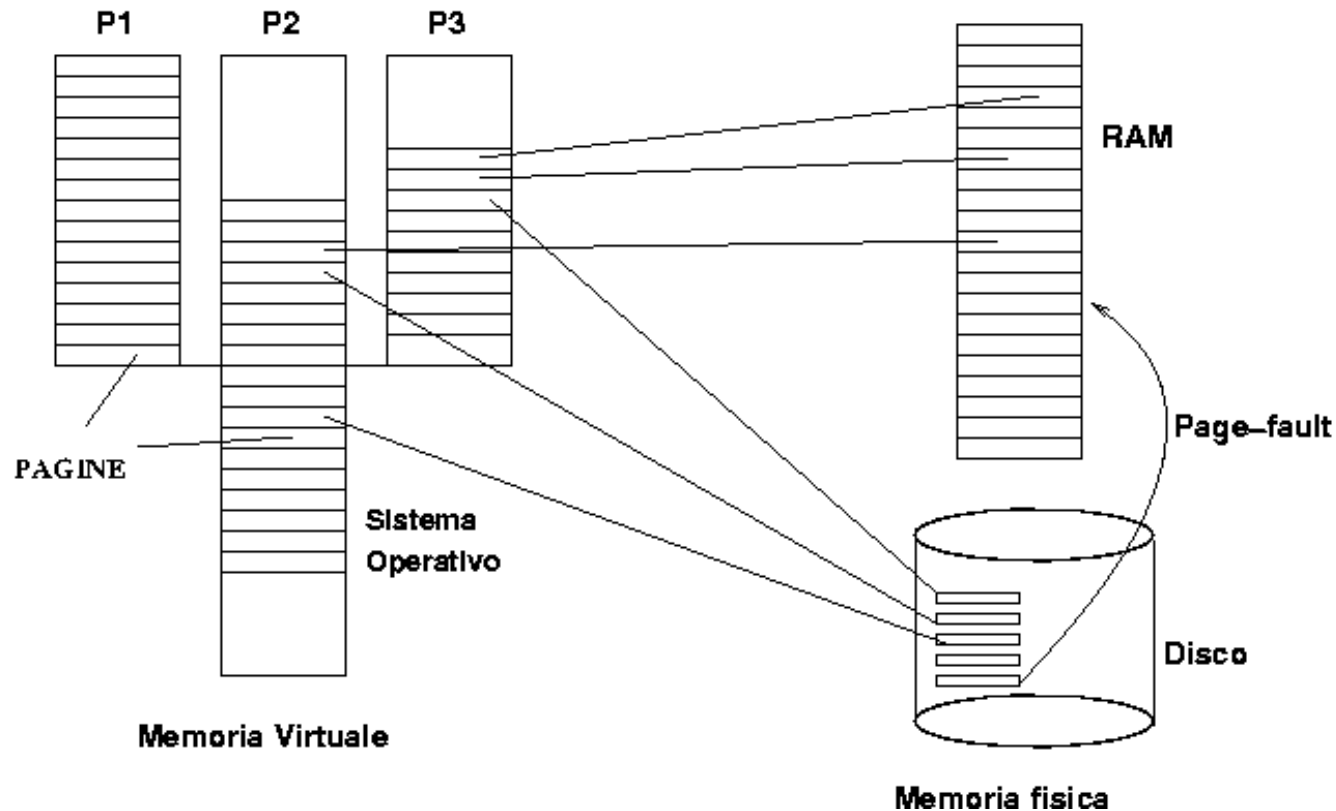
2^{32} byte = 4GByte,

di cui generalmente:

- 2Gbyte sono a disposizione del processo per programmi e dati,
 - gli altri 2Gbyte sono riservati al sistema operativo e sono condivisi da tutti i processi.
- Ogni processo lavora all’interno del proprio spazio virtuale, lasciando al sistema operativo il compito di associare questo spazio virtuale ad uno spazio fisico.

Memoria Virtuale

- se ad un certo istante il processo tenta di accedere ad una istruzione o a un dato di una pagina virtuale mappata su disco fisso => il SO caricherà la pagina da disco a RAM (“Page-fault”).





Memoria di massa

IL GESTORE DELLE INFORMAZIONI

I FILE



- I file sono risorse astratte per conservare informazioni in modo permanente su memoria di massa
- Un file può conservare informazioni di vario tipo e si possono classificare in:
 - file di DATI:
 - file di Programmi che contengono programmi sorgente, oggetto o eseguibili
 - file Speciali, come i file di Sistema o per astrarre dispositivi
 - file Directory che descrivono altri file
- Inoltre i File si distinguono in:
 - File di Record: in cui le informazioni sono logicamente suddivise e in Record e quindi in campi
 - File binari (Byte Stream): sequenze di byte
- I File sono gestiti dal File System

I FILE (2)



- La memoria secondaria permette la memorizzazione persistente di un elevato numero di programmi e dati sotto forma di file.
- Esistono file che sono accessibili in sola modalità di lettura, in sola scrittura o in entrambe le modalità; questa modalità di accesso può essere obbligata dal tipo di dispositivo (si pensi al CD-ROM).
- Ad ogni file è associato un insieme di **attributi** utili nella gestione e nell'accesso dell'archivio. I principali attributi sono:
 - **Identificatore**: È una etichetta unica (di solito un numero) che identifica il file nell'archivio.
 - **Nome**: nome simbolico per riferirsi al file in una forma umanamente leggibile.
 - **Locazione**: È un puntatore alla locazione fisica del file nel dispositivo.
 - **Dimensione**: Dimensione corrente del file (espressa in byte o blocchi).
 - **Data** e **Ora** di creazione e di ultima modifica del file.
 - **Proprietà e protezione**: Informazioni per il controllo degli accessi al file.

I FILE (2)



- Il nome del file generalmente è composto da 2 parti separate da punto: **Nome.Estensione** ad Esempio **MIOFILE.TXT**
- Standard del DOS: 8+3.

I FILE (3)

- Le estensioni più comuni sono:

TIPO DI FILE	ESTENSIONE	PROGRAMMA
File di testo	txt	Editor (Notepad, pico)
Documenti Office	doc ppt xls	Word, power point, Excel.
Formati grafici	jpg gif	Browser Web, editor grafici
Documenti Web	htm html	Browser Web
Acrobat, Postscript	pdf ps	Acrobat reader, ..
Audio	wav au	RealAudio, WMPlayer,..
Sorgenti C e C++	c cpp	Visual Studio,
Matlab	m	Matlab
Sorgenti Java	java	Java Editor
Configurazione	conf (unix)	Editor
Libreria	a so lib dll	
Eseguibile	exe com o nessuna	
Oggetto	obj o	

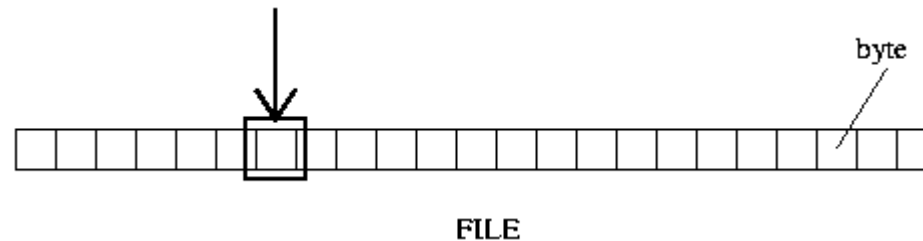
I FILE

- Operazioni principali sui file:

	In DOS	In Linux
Copiare	copy	CP
Spostare	move	Mv
Cancellare	del	Rm
Rinominare	ren	Mv
Visualizzare	type	Cat
Stampare	print	lpr
Creare/modificare	edit	Vari editor

I FILE

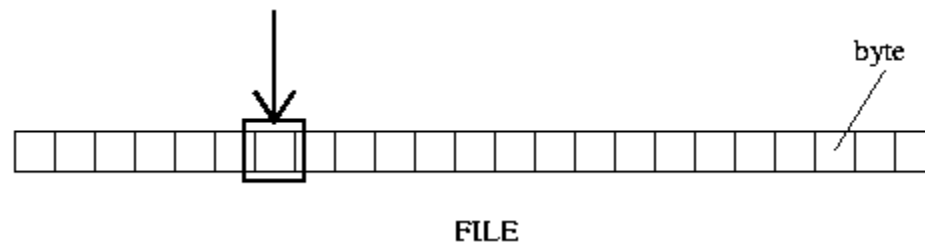
- Il sistema operativo ha il compito di fornire all'utente l'astrazione del file, occupandosi della sua realizzazione fisica con i blocchi del disco. Inoltre il sistema operativo deve fornire un insieme di operazioni per l'accesso al file.
- L'accesso al file in lettura o scrittura avviene mediante un "puntatore" che punta al prossimo byte da leggere o scrivere.



I FILE

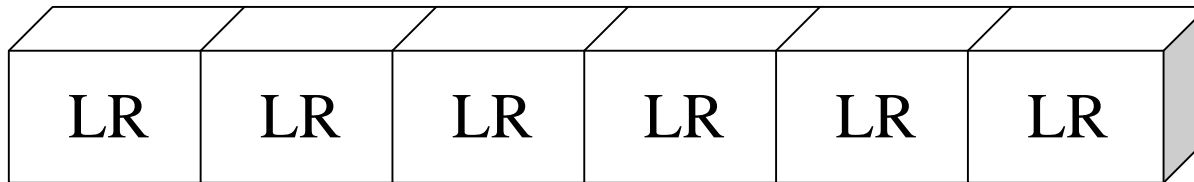
Tipi di Accesso ai FILE:

- accesso "sequenziale":
 - Ad ogni operazione di accesso al file il puntatore si sposta automaticamente in avanti di una posizione. Ad Esempio il nastro magnetico permette solo l'accesso sequenziale
- accesso "diretto" (o "casuale"):
 - il puntatore si può spostare arbitrariamente in punti diversi de l file (riposizionamento). Ad esempio il disco magnetico consente di accedere ai dati in modo arbitrario (spostando opportunamente la testina di lettura/scrittura).



ORGANIZZAZIONE AD ACCESSO DIRETTO

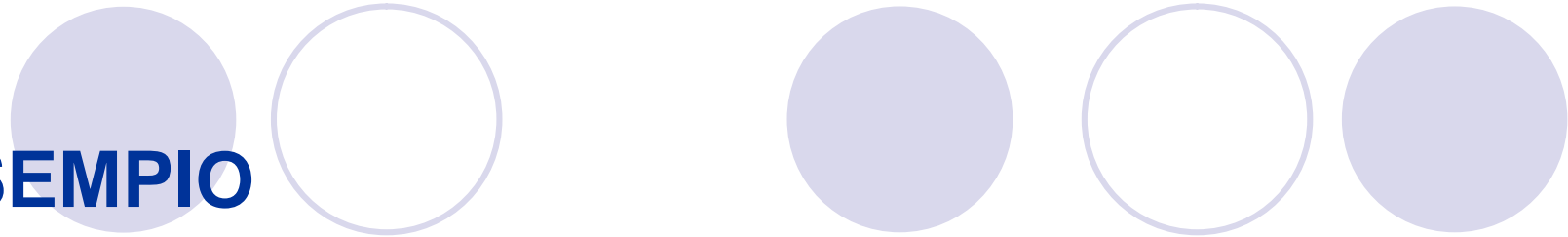
Tutti i record hanno lunghezza fissa



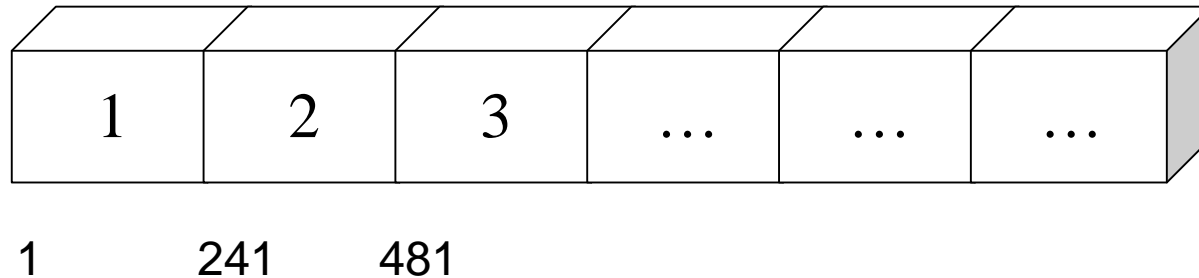
$$\text{RECORD} = (\text{NR}-1) \times \text{LR} + 1$$

Si può dunque accedere direttamente al record che interessa

ESEMPIO



LR= 240 byte



Per leggere il record 3 devo iniziare:

$$(NR-1) \times LR + 1 = (3-1) \times 240 + 1 = 481$$

Si può dunque accedere direttamente al record che interessa ⁵⁷

LE DIRECTORY



- Il numero di files che devono essere memorizzati può essere estremamente elevato => L'idea è quella di raggruppare i files in sottoinsiemi seguendo criteri logici.
- Una *directory* (o cartella) è una scatola che può contenere files o altre directories.
- Tramite le directory i file possono essere memorizzati in una struttura organizzata che generalmente è un albero ad altezza arbitraria.
- Ogni file è collocato in una particolare directory dell'albero ed è identificato univocamente dal nome del file e dal percorso per raggiungere la directory che lo contiene a partire dalla radice dell'albero (root directory).

Directory (path assoluto e relativo)

Nei file-system Unix:

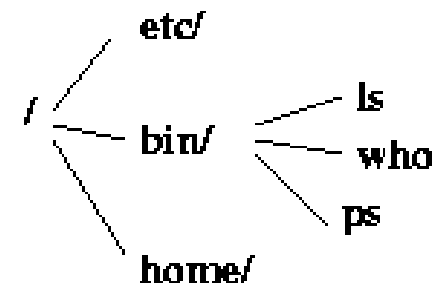
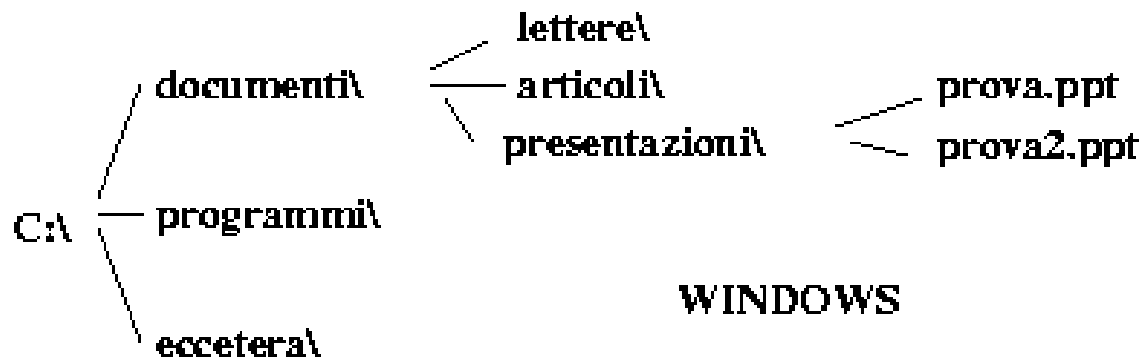
/percorso/directory/nomefile.estensione

Esempio: /bin/ls

Nei file system di Windows:

dispositivo:\percorso\directory\nomefile.estensione

Esempio: C:\documenti\presentazioni\prova.ppt



PARTIZIONAMENTO DI UN DISCO

- Un disco fisso può essere suddiviso in diversi dischi logici denominate “Partizioni” (da 1 a 4).
- Una partizione può essere **Primaria**, **Nascosta** o **Estesa**
- Ogni partizione **Primaria** è vista dal SO come un disco separato
- Se ci sono più partizioni primarie solo una è **Attiva**, mentre le altre vengono nascoste. La partizione attiva è quella in cui il BIOS cerca il Boot Sector
- Una partizione **Estesa** (solo una nel disco) può essere ulteriormente suddivisa in **Unità Logiche** (numero qualsiasi)

PARTIZIONAMENTO DI UN DISCO

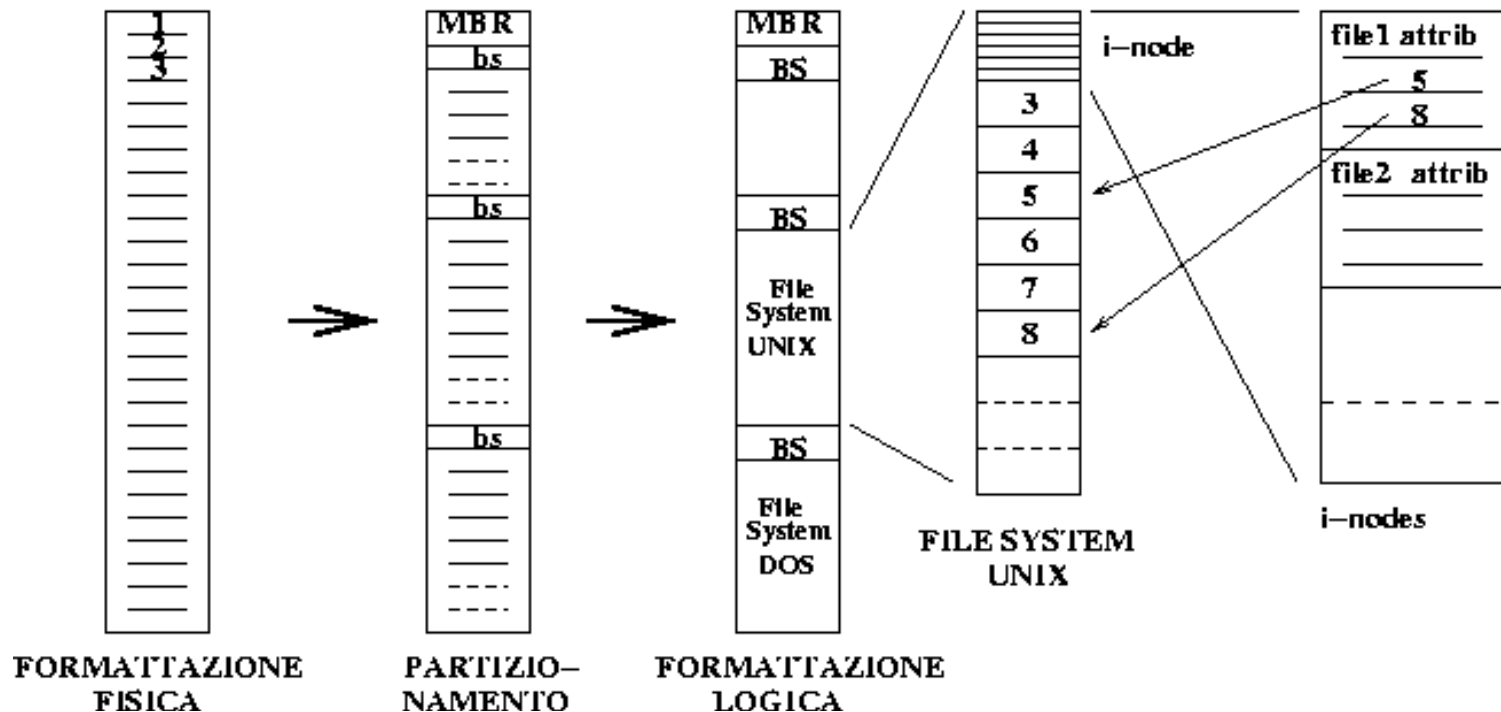
- Il primo settore del Hard Disk (cilindro 0, traccia 0, settore 0) è denominato **Master Boot Record (MBR)** e contiene:
 - La **Partition Table** (Tabella di Partizionamento): tabella di 4 elementi contenente per ogni partizione:
 - Il nome della partizione
 - L'indicazione se è attiva o meno
 - La locazione di Inizio e fine (traccia, cilindro e settore)
 - La Dimensione (come numero di settori)
 - Un programma (**BOOT LOADER**) che permette di avviare il caricamento del sistema Operativo dalla partizione attiva
- mentre il primo blocco di ogni partizione è denominato "**Boot Sector**".
- Il partizionamento può essere fatto tramite comandi (esempio FDISK) o particolari programmi (esempio Partition Magic)

BOOTSTRAP del SISTEMA

- Al momento dell'accensione del calcolatore avviene l'operazione di avvio (bootstrap). Essa costa di 3 fasi:
 - Sequenza di pre-boot o **POST** (Power on Self Test): test sull'hardware presente.
 - Viene eseguito il programma **BOOT LOADER** (presente nel Master Boot Record) il quale, servendosi della tabella delle partizioni, individua la partizione attiva e ne carica in memoria il **Boot Sector**
 - Esecuzione del **BOOT SECTOR (OS LOADER)**: provvede a caricare nella RAM il nucleo del sistema operativo

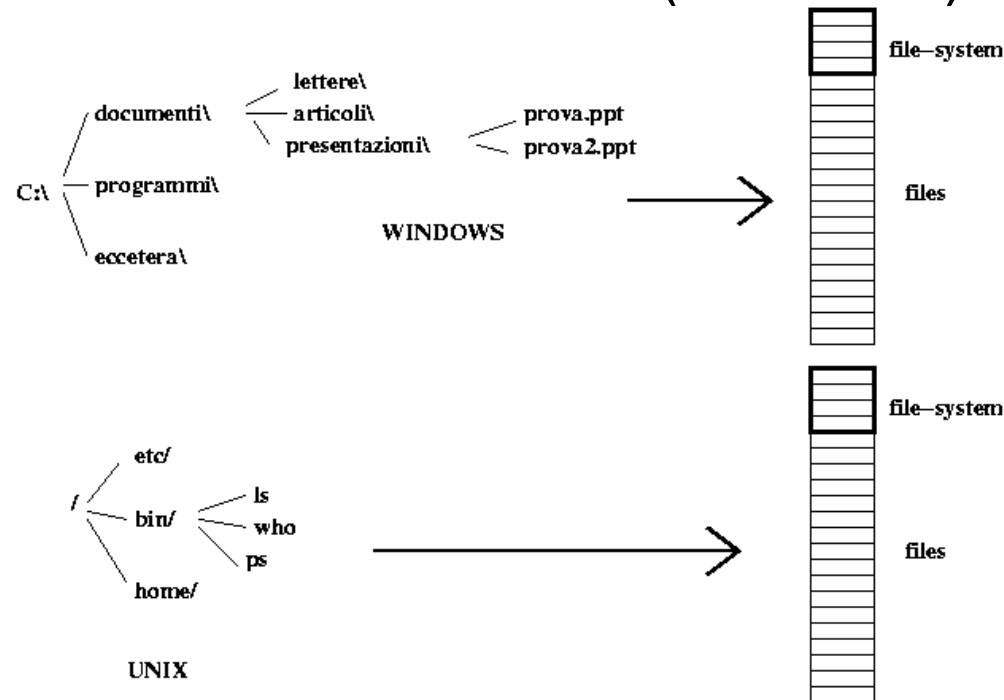
Quindi sull'Hard Disk...

1. il fabbricante effettua la formattazione Fisica (a basso livello) (suddivisione in tracce e settori). Dimensione tipica di un settore è 512 byte.
2. L'utente effettua il partizionamento
3. La formattazione logica: ogni partizione dev'essere partizionata logicamente, creando così il file system.



File System

- Il FS ha lo scopo di fornire all'utente un'interfaccia software per l'uso dei dati su memoria di massa esonerandolo dai problemi di allocazione fisica e da quelli legati alle caratteristiche dei particolari supporti fisici
- Il FS consente all'utente di riferirsi ai file attraverso nomi simbolici (FS Logico) senza preoccuparsi della gestione e della registrazione fisica dei dati (FS Fisico)



Tipi di File System



- Esistono diversi tipi di file-system.
- Ad esempio il sistema operativo Windows può utilizzare dischi con FAT o con NTFS.
- FAT è un file-system per sistemi operativi "monoutente" poiché non prevede tra gli attributi dei file l'attributo "proprietario".
- Con l'introduzione di sistemi operativi Windows multiutente (WindowsNT, Windows2000, ecc) è stato necessario introdurre un nuovo tipo di file-system (NTFS) multiutente in cui tra gli attributi del file sono stati aggiunti il "proprietario" e i permessi di accesso.
- Unix e linux utilizzano file system basati sugli i-node. L'inode è una struttura dati che contiene tutti gli attributi di un file o di una directory. Ovviamente anche i file-system per Unix sono di tipo multiutente

File System: principali funzioni

- Gestire in modo ottimale lo spazio disponibile nel disco
- Garantire all'utente l'accesso al file in maniera veloce, anche in presenza di richieste di accesso da parte di altri utenti
- Fornire agli utenti meccanismi di protezione dei file, rispetto a interventi dannosi o non autorizzati da parte di altri utenti
- Fornire operazione di uso comune sui file:
 - Cancellazione del file
 - Rinominazione
 - Elenco di file
 - Visualizzazione di informazioni di un file (spazio occupato, data di creazione, protezioni)

File System: principali funzioni

- Il FS consente di riferirsi alle informazioni registrate avvalendosi di identificatori che sono registrate in tabelle dette DIRECTORY.
- Ogni riga della Directory è un descrittore di un file e contiene le seguenti informazioni:
 - Il Nome del file
 - La data di creazione
 - Il numero di record contenuti
 - La dimensione del record
 - Le modalità consentite per l'accesso (esecuzione, lettura, scrittura) differenziato per i vari utenti

File System FISICO



- A livello fisico i File vengono memorizzati come sequenze di Blocchi.
- Un Blocco è una unità di informazione che può essere trasferita con un'unica operazione di lettura/scrittura e corrisponde a un settore o Multiplo di settori

File System FISICO

- Ogni dispositivo di memorizzazione contiene almeno:
 - il Descrittore del sistema di archiviazione che può contenere:
 - l'etichetta del disco
 - La lunghezza massima dei file
 - 1° blocco usato per i file e il numero di blocchi presenti. Ecc...
 - Tabella dei descrittori: ogni riga contiene un descrittore di un file ossia informazioni del tipo: nome del file, lunghezza del record e n° di record, data di creazione del file, data ultima modifica, ecc...
 - Directory: l'elenco dei file presenti è mantenuto in essa.
 - La FAT o la Mappa dei blocchi

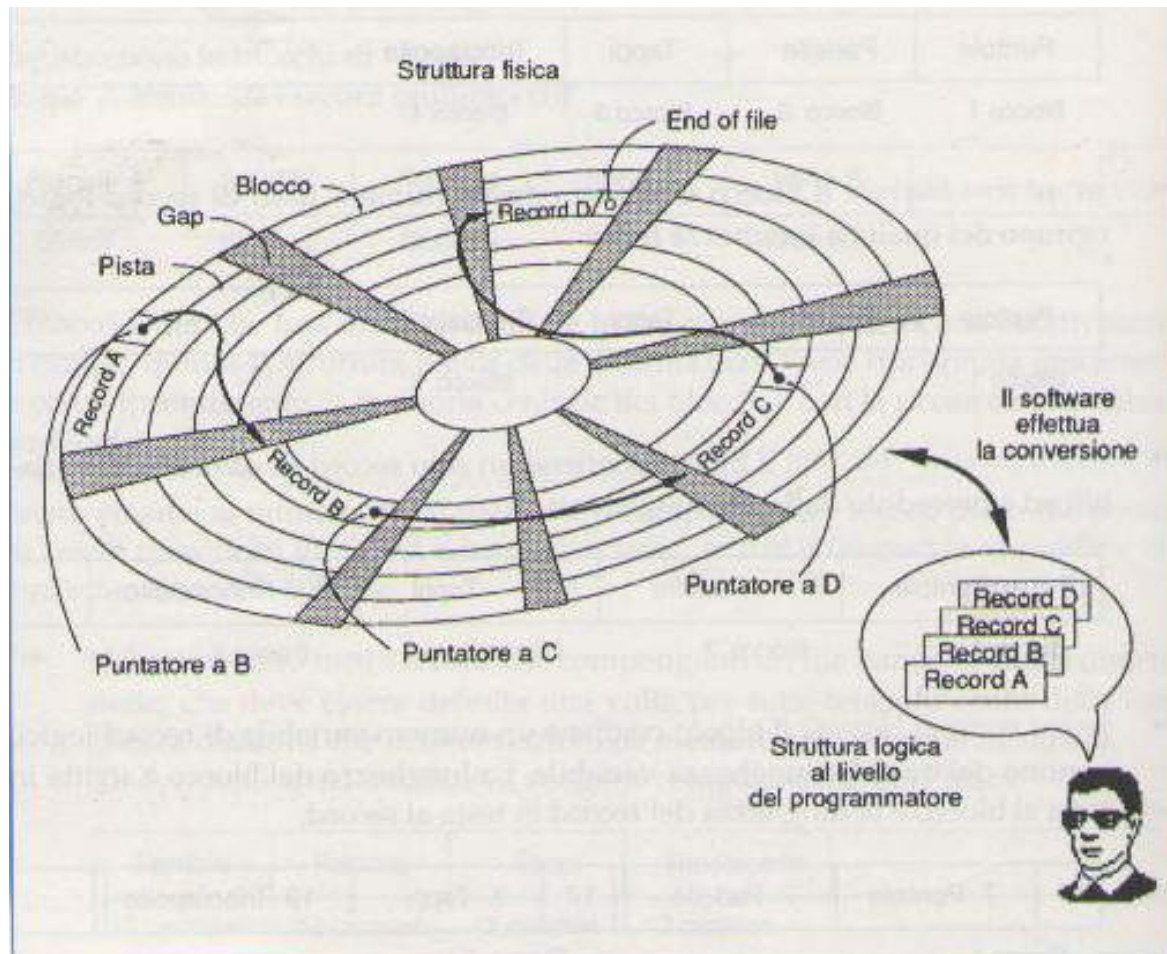
File System FISICO



- Il fs Fisico si occupa:
 - Di “trasformare” i Record Logici in Record Fisici (insieme di blocchi)
 - della memorizzazione dei file usando diversi **Metodi di ALLOCAZIONE** dei blocchi, contigui o no (normalmente non contigui). Le tecniche principali sono:
 - Contigua
 - Concatenata
 - Tabella di puntatori (FAT)
 - Indicizzata (mappa dei blocchi)
 - Occorre gestire anche i blocchi liberi (normalmente concatenati in una lista dello Spazio Libero)

File System FISICO: esempio1

I Record logici possono essere ad esempio i dati dei dipendenti (nome, cognome, stipendio, ecc...). Quindi il record logico visto dal programmatore è cosa diversa da come lo vede il file system:

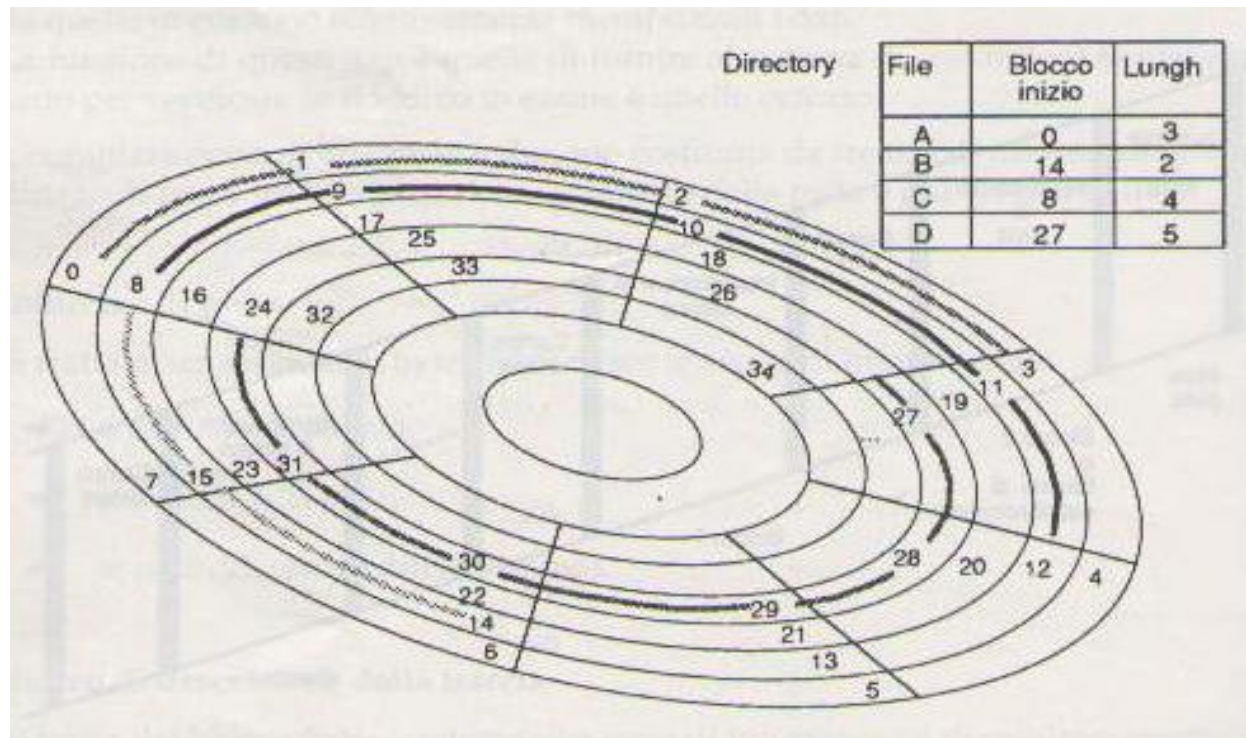


OSS: i record logici sono contigui, quelli fisici no

Allocazione contigua

I file sono allocati in blocchi contigui.

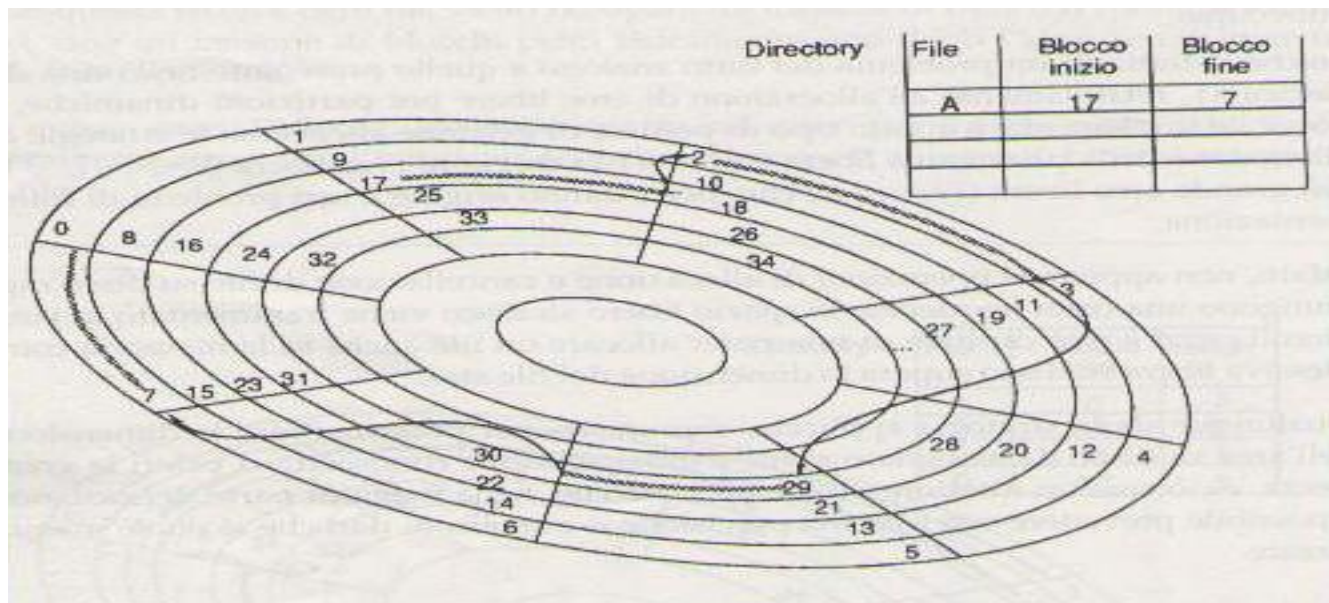
- Vantaggi: pochi movimenti della testina per visitare i blocchi di un file
- Svantaggi:
 - trovare lo spazio più idoneo per ogni nuovo file
 - Eccessiva frammentazione



Allocazione concatenata

I file sono allocati in una lista di blocchi che possono essere sparsi ovunque nel disco. La directory, oltre alle informazioni sul file, conterrà un puntatore al 1° e all'ultimo blocco del file.

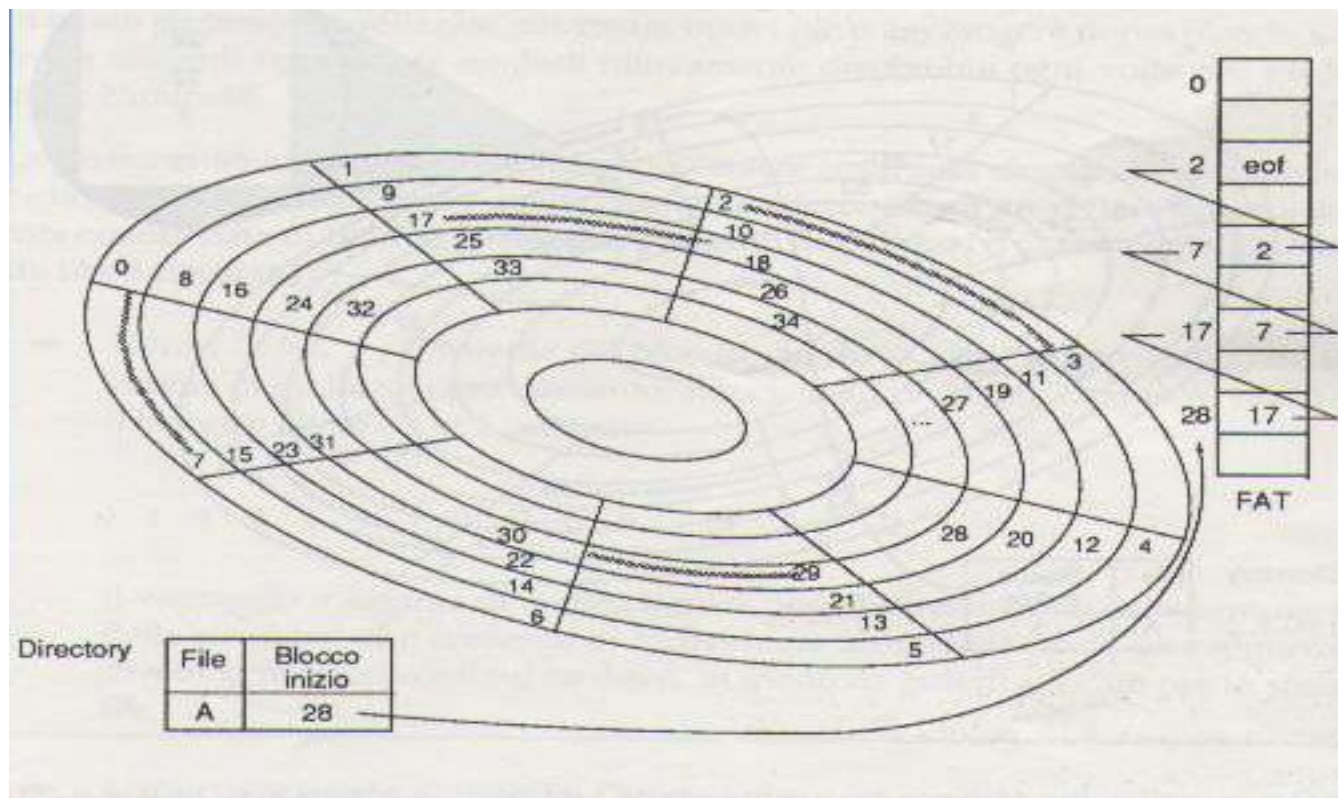
- Vantaggi: si elimina il problema della frammentazione
- Svantaggi:
 - Affidabilità: basta perdere un solo puntatore per non riuscire ad accedere al file



Allocazione a FAT

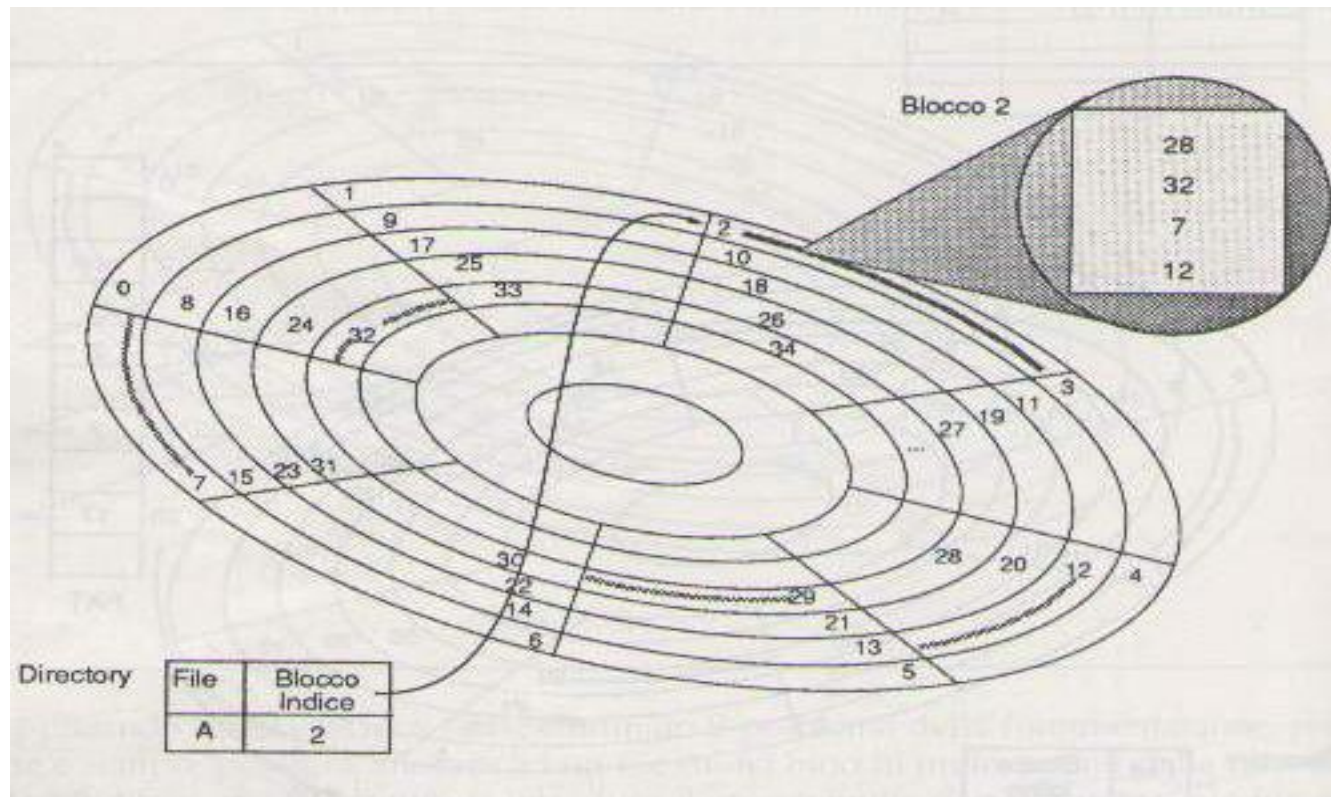
Si usa una tabella formata da tanti elementi quanti sono i blocchi su disco. **Ogni elemento della FAT contiene:**

- Un puntatore al blocco successivo (se il blocco appartiene a un file)
- Una indicazione se è libero (blocco non usato)



Allocazione a Mappa di blocchi

Si usa una tabella per ogni file, contenente la sequenza dei puntatori ai blocchi del file. La tabella può essere memorizzata nella riga della Directory associata al file oppure in un blocco a se stante



Riassumendo

A decorative graphic consisting of two rows of circles. The top row has two circles: a solid light purple one on the left and an outlined light purple one on the right. The bottom row has three circles: a solid light purple one on the left, an outlined light purple one in the middle, and a solid light purple one on the right.

- Il file system:

- All'utente offre:

- un'organizzazione logica ad albero di Directories all'interno delle quali ci stanno file e altre Directories
 - Una serie di operazioni per lavorare facilmente sui file

- da un punto di vista fisico:

- alloca i file utilizzando i blocchi nel disco
 - e ne tiene conto tramite Fat o Mappa di blocchi

File System Logico



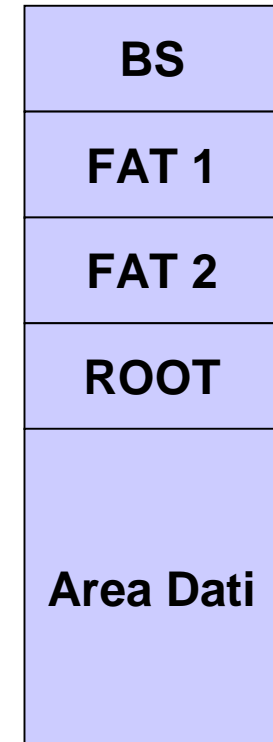
- Permette all'utente la visione logica ad albero
- Col FS logico i descrittori dei file vengono raggruppati in più directory => vantaggi:
 - vengono caricati in memoria solo le directory relativi ai file che si stanno usando (si evita spreco di memoria)
 - Si possono raggruppare nella stessa Directory file tra loro correlati (es. file di un utente):
 - Si può dare lo stesso nome a file che si trovano in directory diverse
- I SO Windows, DOS, Unix usano un sistema ad albero

Un caso reale: Il File System FAT

Descrizione del Sistema di archiviazione

Il disco è visto logicamente in 2 aree:

- **Area di Sistema**, contenente:
 - **L'Area Riservata**: a sua volta contiene
 - il boot sector (nel 1° settore della partizione)
 - Parametri che definiscono la lunghezza dei Settori, il numero di settori x Cluster, ecc...
 - la **FAT (File Allocation Table)**: 2 copie di questa tabella, una di riserva;
 - la **DIRECTORY principale (radice)**
- **Area Dati**: area realmente utilizzata per la memorizzazione dei dati.



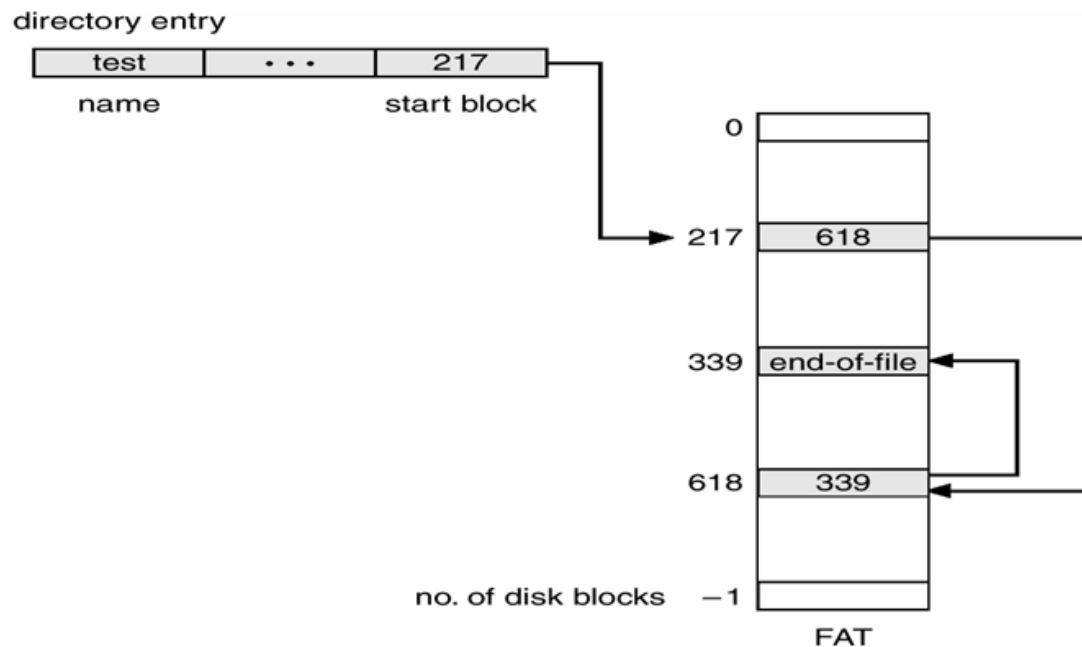


Il File System FAT: Area Dati

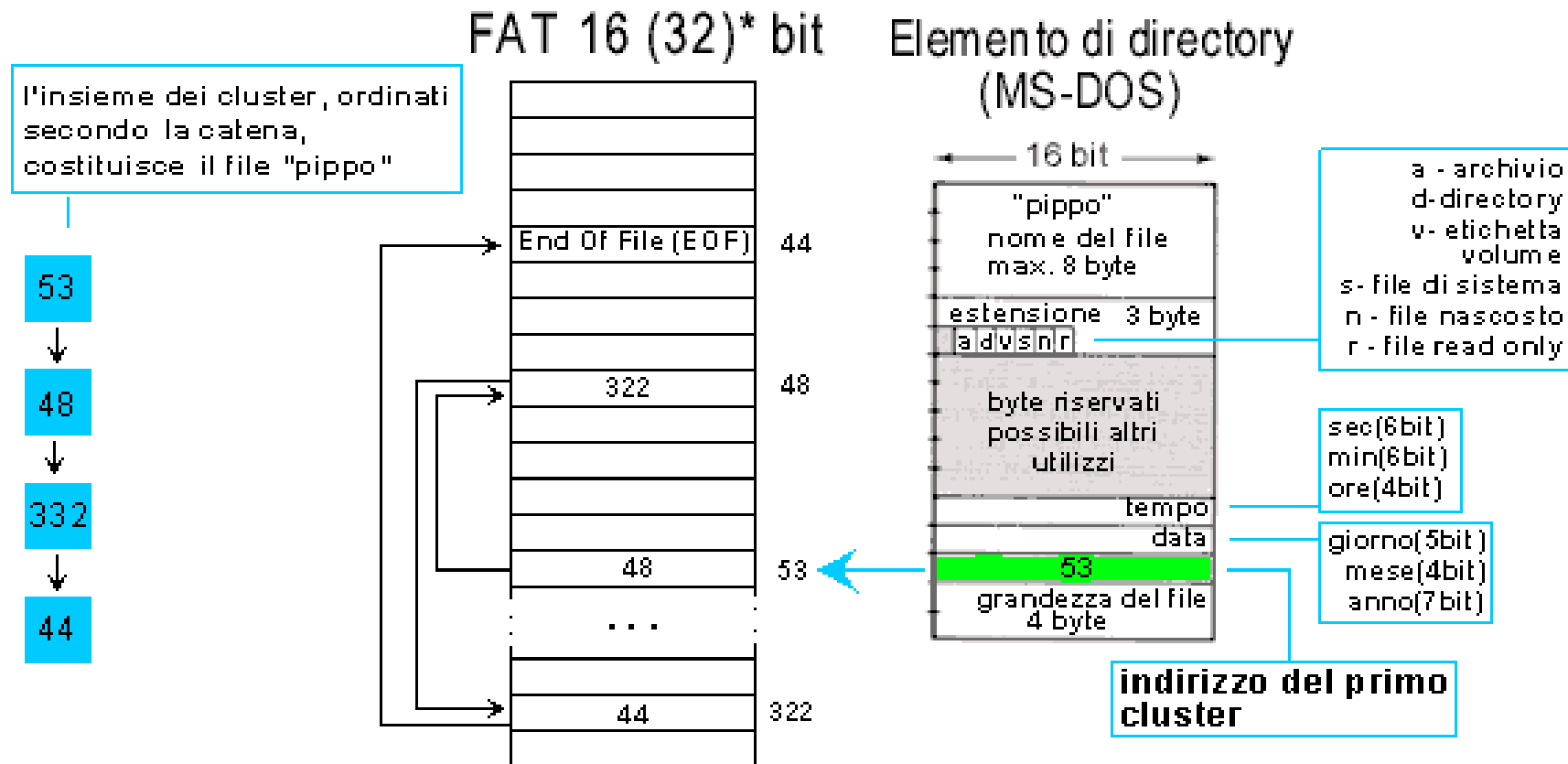
- I settori vengono individuati da un numero di settore logico, a partire da zero => primo settore logico = Settore 0
- Un settore ha dim. 512 byte (1/2 kb)
- I dati e i programmi vengono registrati come file non necessariamente in modo contiguo sulla superficie del disco (ad esempio come avviene per la musica incisa nei dischi musicali)
- L'allocazione su disco avviene raggruppando i settori in **CLUSTER**.
- I file vengono allocati nell'Area Dati per cluster. Ogni file occupa almeno un Cluster.

FAT (File Allocation Table)

- La FAT registra l'assegnazione dei cluster i file
- Per maggior sicurezza in ogni partizione ci sono 2 copie di FAT
- La FAT è una tabella in cui ogni elemento corrisponde a un cluster. Esso è un numero che può rappresentare:
 - L'indirizzo del cluster successivo;
 - Un valore speciale per rappresentare la fine del file, oppure se è inutilizzato o se danneggiato



Altra visione della FAT



FAT 12, 16, 32

- Il nome deriva dalla grandezza degli elementi della tabella:
- FAT12 (usata nei floppy):
 - ogni elemento è 12 bit,
 - Con 12 bit => si possono memorizzare valori da 0 ÷ 4095 => fino a 4095 cluster.
 - Nei floppy ogni cluster contiene un solo settore => $4095 \times 512 = 2\text{Mb}$
- FAT16:
 - Utilizza cluster con almeno 4 settori (2kb) => può gestire dischi fino a 128MB ($2^{16} \times 2\text{kb}$).
 - Ogni volta che si raddoppia la capacità del disco => viene raddoppiata la dimensione dei Cluster fino a un max di 32Kb per Cluster => dischi max 2Gb ($2^{16} \times 32\text{kb}$).
 - Crescendo la dim. Del cluster si spreca spazio con i file piccoli.
- FAT32:
 - Risolve il problema dello spazio usando cluster di 4kb
 - Pur essendo la tabella a 32 bit, 4 bit sono riservati => ne usa 28.
 - Con 28 bit si possono gestire 2^{28} cluster => fino al limite di 1024 Gb ($2^{28} \times 2^2 \times 2^{10}$)
- La FAT32 è disponibile a partire da Windows95 OSR2.

La Directory nel File System FAT

- È una tabella in cui ogni elemento contiene le informazioni relative a un file o ad una subdirectory
- Ogni directory e subdirectory contiene 2 speciali riferimenti:
 - . Riferimento alla directory stessa
 - .. Puntatore alla directory genitore
- Ogni elemento è lungo 32 byte e contiene varie informazioni:
 - Nome corto: 8 byte per il nome e 3 per l'estensione
 - Attributi: 1 byte; questi bit danno informazioni sul file (se è nascosto, di sistema, sola lettura, ecc...)
 - Orario dell'ultima modifica
 - Data dell'ultima modifica
 - Numero del 1° Cluster per quel file
 - Dimensione del file



Il file System di UNIX

Il file System di UNIX

- In UNIX il termine file system è usato per indicare ogni partizione del disco
- Ogni file system è gestito per mezzo di un descrittore del sistema di archiviazione chiamato **SUPERBLOCCO**.
- Ogni file è descritto da un descrittore, chiamato **I-NODE** (nodi indice), che contiene le informazioni relative al file (non il nome del file)
- L'area contenente gli i-node si chiama **I-LIST** (lista degli indici)
- L'associazione tra nome del file e i-node compare nella **Directory** (organizzata ad albero)
- Il FS contiene anche i puntatori per la gestione dei blocchi liberi (free list)



Il file System di UNIX: Superblocco

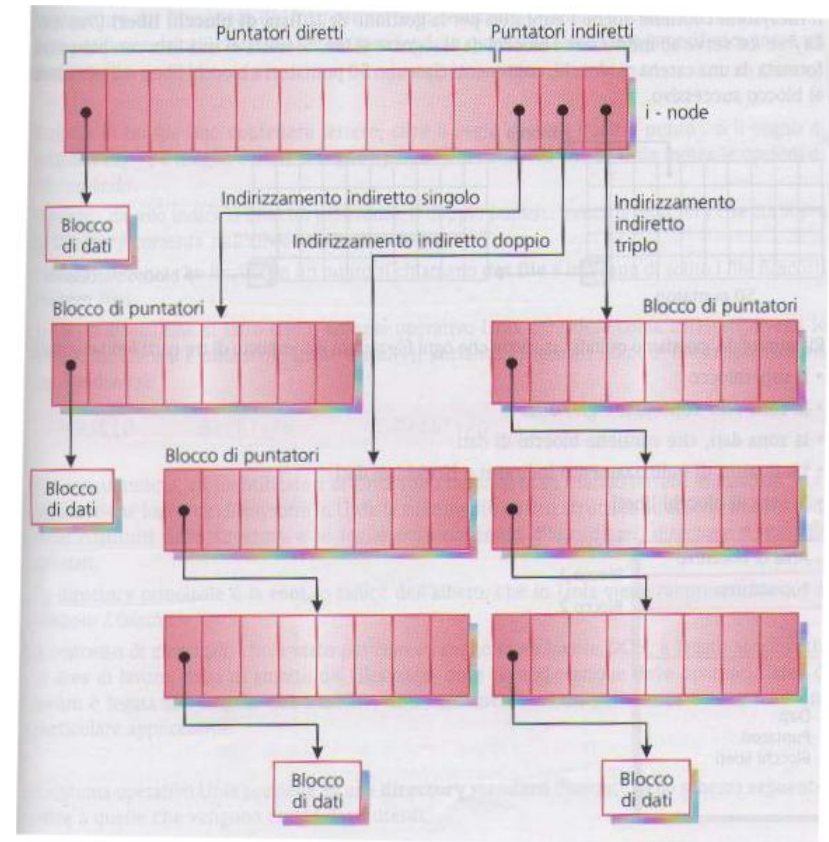
- Il **SUPERBLOCCO** descrive lo stato complessivo del file system
- È creato al momento della creazione del FS stesso e contiene i parametri:
 - Dimensione del FS
 - Dimensione dell'area per i-list
 - Dimensione di un blocco
 - Numero di blocchi per i dati
 - Numero di i-node
 - Numero del 1° i-node del FS corrispondente alla Directory Root (/)
- Per evitare perdita di dati il SuperBlocco viene duplicato in altre posizioni all'interno del FS

Il file System di UNIX: I-NODE

- Ogni file è descritto da un i-node nella tabella degli i-node
- Ogni i-node è identificato da un numero (i-number)
- La Directory associa a ogni file un i-node attraverso il numero
- l'i-node è formato da 64 byte in cui:
 - Nella prima parte ci sono informazioni sui file quali:
 - Tipo di file (normale, speciale, directory)
 - Permessi (lettura, scrittura, esecuzione)
 - Dimensione del file
 - Nome del proprietario (UID e GID)
 - Delle date (ultimo accesso, ultima modifica, di creazione)

Il file System di UNIX: I-NODE

- Nella Seconda parte dell'i-node vi è un vettore di 13 elementi (in Riferimento al FS ext2). questi elementi contengono puntatori:
 - I primi 10 elementi puntano a 10 blocchi di dati (Puntatori Diretti)
 - l'11° punta a un blocco che contiene puntatori (Indirizzamento indiretto singolo)
 - Il 12° punta a un blocco che punta ad altri blocchi di puntatori (Indirizzamento indiretto doppio)
 - Il 13° indirizzameno indiretto triplo (per file molto grandi)

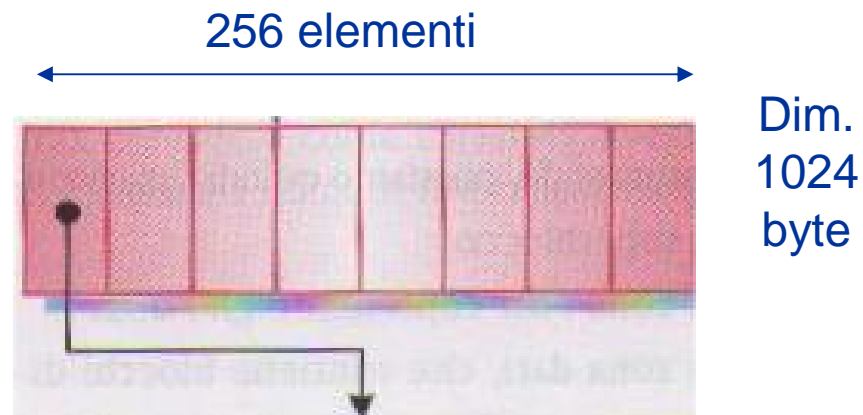


I-NODE



In riferimento al FS ext2, consideriamo:

- Blocchi di dim. 1024 byte ciascuno
- nei blocchi usati come puntatori, ogni riferimento occupa 4 byte => 256 puntatori in ogni blocco.



I-NODE: ESEMPIO

- con il solo indirizzamento diretto qual è la dim. Massima di un file?

I primi 10 puntatori puntano a 10 blocchi di dati, ogni blocco è 1024 byte => **10 x 1024 = 10.240 byte**. (10 Kb)

- con il 1° indirizzamento indiretto?

Se il file è grande l'11° puntatore punta a un blocco con 256 puntatori a blocchi di dati

=> $10 \times 1024 + 256 \times 1024 = 272.384 \text{ byte}$ (270 Kb)

- con il 2° indirizzamento indiretto?

il 12° indirizza un blocco con 256 puntatori a blocchi che contengono a loro volta puntatori a blocchi di dati

$10 \times 1024 + 256 \times 1024 + 256 \times 256 \times 1024 = 67.381.248 \text{ byte}$ (67 Mb)

- con il 3° indirizzamento indiretto?

=> $10 \times 1024 + 256 \times 1024 + 256 \times 256 \times 1024 + 256 \times 256 \times 256 \times 1024 = 17.316.980.736$
byte (17 Gb)

Altri File System



- I file system per i CDROM (ISO 9660):
 - Windows e Linux utilizzano lo stesso file system per i CDROM => i CD si possono così utilizzare su entrambi i sistemi
 - Inizialmente il 1° standard fu **ISO 9660**, con la limitazione dei nomi corti (8+3)
 - poi ci fu il supplemento **Joliet**: nomi fino a 64 caratteri). Questo però era piuttosto complesso
 - Oggi sia Windows che Linux usano **UDF** (Universal Disk Format)

I File System di Windows

- I file system di Windows sono:
 - FAT con le sue estensioni (Visti prima)
 - NTFS (New Technology File System):
 - Nomi lunghi fino a 255 caratteri
 - Offre funzioni di protezione: è progettato per offrire protezione e restrizioni di accesso a livello di file
 - Inoltre offre Compressione e Crittografia dei file

I File System di Linux



- I file system di Linux sono:
 - Minix
 - ext (Extended file system)
 - Attualmente ext2 e ext3
- Linux supporta anche FAT del dos => riesce ad accedere ai suoi file
- Nomi lunghi 255 caratteri e Case Sensitive

I File System di Linux: tipi di file

- 3 tipi di file:
 - Normali
 - Directory
 - Speciali: rappresentano i dispositivi di I/O
- In linux tutto è visto come file => se ad esempio voglio accedere al CDRom, lo monto in un'apposita cartella (`\etc\cdrom`) e lo tratto come un qualsiasi file

Organizzazione logica in UNIX

- Ad ogni utente è assegnata una directory personale detta **Home Directory**
- Il *pathname* (o nome del percorso) di un file è la sequenza di *directories* che si deve attraversare per raggiungere quel determinato file a partire dalla radice dell'albero:
 - *Pathname assoluto*: intero cammino descritto a partire dalla root. Es. /home/fabio/documenti/prova.txt
 - *Pathname relativo*: il cammino parte dalla directory corrente. Se ad esempio siamo già in directory home potremmo scrivere: fabio/documenti/prova.txt

Organizzazione logica in UNIX

- Con riferimento ai file, gli utenti del sistema sono classificati secondo 3 tipologie:
 - Proprietario (owner, user), la persona che ha creato il file
 - Gruppo
 - Others (altri)
- Esiste un utente particolare: SuperUser, il Dio del sistema

Organizzazione logica in UNIX

- Per ogni file ci sono 3 tipi di autorizzazioni relative a ciascun tipo di utente => un totale di 9 autorizzazioni:
 - R (read, lettura)
 - W (write, scrittura)
 - X (esecuzione, execute)
 - Ognuna dev'essere specificata al tipo di utente:

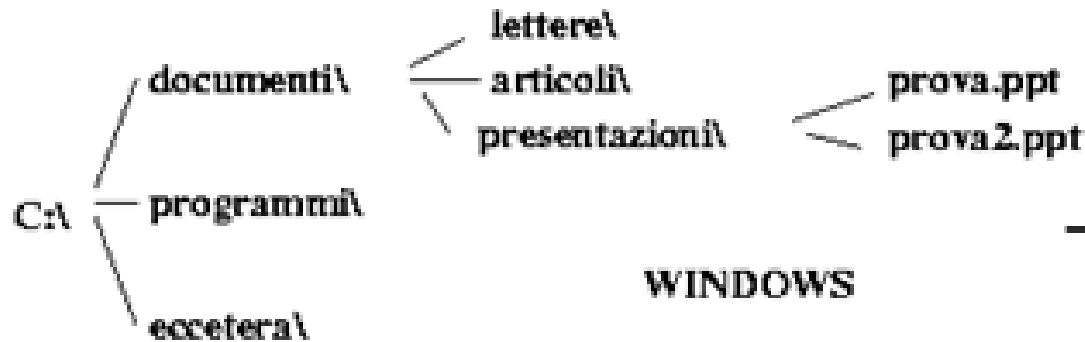


esempio

```
Telnet - 192.168.0.98
Connessione Modifica [Esc] Invia [Inv]
[root@infserver root]# ls -l /usr
Totale 136
drwxr-xr-x  3 root  root    36864 ott  1 11:01 bin
drwxr-xr-x  2 root  root    4096 Feb  6 1996 dict
drwxr-xr-x  2 root  root    4096 Feb  6 1996 etc
drwxr-xr-x  2 root  root    4096 set 17 12:30 games
drwxr-xr-x 93 root  root    8192 ott  1 11:01 include
drwxr-xr-x  3 root  root    4096 ott 30 12:31 java
drwxr-xr-x  8 root  root    4096 set 17 13:06 kerberos
drwxr-xr-x 66 root  root   45056 ott  1 10:46 lib
drwxr-xr-x  7 root  root    4096 ott  1 11:01 libexec
drwxr-xr-x 13 root  root    4096 ott 30 12:32 local
drwxr-xr-x  2 root  root    8192 set 17 13:03 sbin
drwxr-xr-x 100 root  root    4096 ott  1 10:46 share
drwxr-xr-x  3 root  root    4096 set 17 13:10 src
lrwxrwxrwx  1 root  root     10 set 17 12:25 tmp -> ../var/tmp
drwxr-xr-x  8 root  root    4096 set 17 13:06 X11R6
[root@infserver root]#
```

Organizzazione Logica in DOS -Windows

- La directory principale è C:\



INTERFACCIA UTENTE



I sistemi operativi sono dotati di una interfaccia che permette la comunicazione con l'utente.

Esistono 2 tipi di interfaccia utente:

- l'interfaccia grafica (GUI)
- e l'interfaccia a linea di comando (shell).

INTERFACCIA UTENTE



Le funzionalità (sia GUI che a shell) sono le stesse, anche se ottenute con modalità diverse:

- Accesso al file system
 - Browsing delle directory
 - Visualizzazione dei file (attributi e contenuto)
 - Creazione/cancellazione file e directory
- Gestione risorse
 - Installazione configurazione driver
 - Gestione dischi (formattazione, installazione file system)
 - Gestione code di stampa
 - Gestione della rete
- Gestione Utenti
- Esecuzione di programmi installati dall'utente
 - Altri programmi di utilità per la gestione del calcolatore
 - Software applicativo

L'Interfaccia linee di comando (SHELL)

- IL dialogo tra utente e macchina avviene mediante un linguaggio elementare composto da un insieme di comandi e relative opzioni.
- L'interfaccia è fondamentalmente un interprete di comandi: la shell presenta all'utente una stringa (Prompt) che rappresenta l'invito all'utente di digitare un comando;
- il comando può essere una funzionalità gestita dalla shell (prova a digitare HELP nella shell DOS) oppure è il nome di un programma che deve essere presente in opportune directory del file system.
- La shell gestisce una tabella di variabili (variabili d'ambiente) attraverso la quale l'utente può personalizzare l'ambiente di lavoro.
- Nella shell DOS e nella shell BASH (Linux) il comando "set" consente di visualizzare e modificare le variabili d'ambiente.
- La variabile d'ambiente più importante è PATH, mediante la quale viene stabilita la lista delle directory in cui cercare i comandi.
- (Esercizio: consultare le variabili d'ambiente e le directory di PATH)
- Un'altra caratteristica importante della shell è la Directory Corrente.

Operazioni principali in Windows/Dos

DOS	UNIX	SIGNIFICATO
DIR	ls	Elenco dei file nella dir corrente
CD newdir	cd newdir	Cambia la dir corrente in modo relativo
CD \dir1\dir2	cd /dir1/dir2	Cambia la dir corrente in modo assoluto
CD ..	cd ..	dir corrente = genitore
MD newdir	mkdir newdir	Crea nuova dir
REN file1 file2	mv file1 file2	Rinomina file1 in file2
TYPE file.txt	less file.txt	Visualizza il contenuto del file di testo
COPY file1 file2	cp file1 file2	Copia file1 in file2
DEL file1	rm file1	Cancella file1

Per i comandi pag. 286-Scorzoni)

Esercizi

- Fare esercizi in DOS utilizzando CD, Dir, Copy e Del
- Fare anche esercizi (con cd) su Path assoluti e relativi

SISTEMI OPERATIVI MULTIUTENTE

- Un sistema operativo è multiutente se è in grado di identificare i diversi utenti che accedono al sistema ed applicare conseguentemente politiche di accesso personalizzate. Gli utenti possono essere associati in gruppi permettendo la definizione di criteri di autorizzazione per tutti i membri del gruppo stesso.
- Il sistema operativo deve per questo avere le seguenti caratteristiche:
 - gestione di una struttura dati per le informazioni degli utenti (dati personali, informazioni per l'autorizzazione, personalizzazioni dell'ambiente di lavoro)
 - gestione di un meccanismo di verifica dell'identità (Esempio Username e Password)
 - utilizzo di un file-system multiutente per poter applicare criteri di accesso diversificate in base all'identità dell'utente e alla sua eventuale appartenenza a gruppi (criteri di lettura/scrittura sui file dati e di esecuzione sui programmi)

Interfaccia Grafica per l'Utente(GUI)

- E' una modalità visuale di interazione con l'utente (active desktop, finestre, sistema di puntamento con mouse).
- L'interazione e' semplificata dall'utilizzo di icone attivabili (mediante il sistema di puntamento) le quali ricordano graficamente la funzione associata.
- Questa modalità e' fortemente integrata nel sistema operativo Windows.
- Per il sistema operativo Unix esistono invece diversi GUI tra cui scegliere; i più utilizzati sono Gnome e KDE