

# INFORMATICA GENERALE

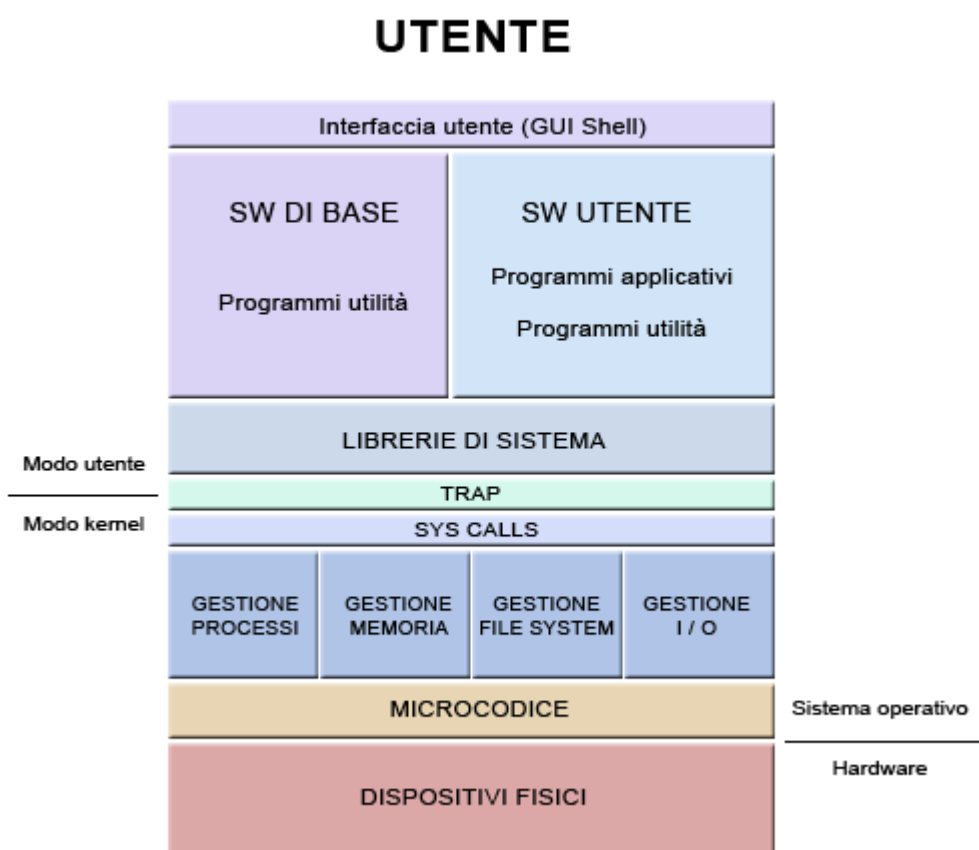
PARTE 2: SOFTWARE

A CURA DEL PROF. CAVAZZA GENNARO

# CLASSIFICAZIONE E PRODUZIONE 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 nasconde la complessità al software applicativo.
- **I 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 (Redhat, Debian, Gentoo, 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.)

## LICENZE D'USO

Il software utilizzato sui calcolatori può essere classificato anche in base al tipo di licenza d'uso. Le licenze regolamentano in vario modo i diritti di utilizzo, redistribuzione, modifica, commercializzazione, accesso alle specifiche esterne (interfaccia), alle specifiche interne (codice sorgente) e proprietà intellettuale.

**Software commerciale:** Il software commerciale è software sviluppato da un'azienda allo scopo di guadagnare dal suo uso. Ad esempio gran parte software Microsoft (Windows, MS Office, Visual Studio, ecc) è commerciale.

**Freeware e Shareware:** Software distribuito senza codice sorgente e che può essere ridistribuito (tipicamente via Internet). Il Freeware non prevede costi nell'utilizzo, mentre Lo Shareware impone un costo per il suo uso dopo un periodo di prova. La licenza Freeware viene molto utilizzata dalle aziende come manovra per vendere software commerciale (spesso denominata "Professional") propagandato mediante la distribuzione Freeware di una versione con caratteristiche limitate (spesso denominata "Light"). Ad esempio Acrobat è software commerciale, mentre "Acrobat Reader" (permette solo la lettura di documenti PDF) è Freeware.

**Open Source:** Esistono diversi tipi di licenze Open Source. Tutte queste licenze hanno caratteristiche comuni: sono permessi l'uso, l'accesso a specifiche e codici sorgenti, mentre possono essere regolamentati proprietà intellettuale, modifiche, redistribuzione e commercializzazione. Le principali licenze Open Source sono:

**GPL** (General Public License) è la licenza più utilizzata per il Software Open Source. La GPL impone restrizioni sulla possibilità di modificare e ridistribuire il software, ovvero proibisce i brevetti proprietari relativi alla modifica del software, e richiede che gli stessi termini di libera distribuzione siano allegati quando si ridistribuisce il software o anche derivazioni di esso.

**LGPL** (Lesser GPL) è una GPL con meno restrizioni riguardo l'inclusione in software non libero. È stata pensata in pratica solo per le librerie, in alcuni casi particolari. Le librerie sotto questa licenza possono essere usate da software con licenza diversa da GPL/LGPL; questo può essere utile, paradossalmente, per promuovere il software Open.

**FDL** (Free Documentation License) applica ai documenti i principi della licenza GPL.

**BSD:** licenza meno restrittiva rispetto alle licenze GNU.

**Free Software / Public Domain:** tutto è permesso senza restrizioni e proprietà intellettuali.

## PRODUZIONE DEL SOFTWARE

La produzione del software richiede diversi strumenti di lavoro che, soprattutto se l'interfaccia con l'utente è di tipo grafico (Graphical User Interface, GUI), possono essere integrati in un unico ambiente multifunzionale detto IDE (Integrated Development Environment). Gli strumenti essenziali sono:

**un editor** per la scrittura del programma sorgente (scritto nel linguaggio di alto livello)

**un programma traduttore** per la traduzione del linguaggio sorgente in un programma eseguibile (scritto in linguaggio macchina)

**un debugger** che serve in fase di test del programma per verificare passo passo il risultato delle singole istruzioni, per individuare e correggere eventuali errori di programmazione. La creazione di un programma richiede generalmente l'iterazione nell'utilizzo di questi strumenti di lavoro eliminando di volta in volta gli errori di programmazione. Questa ricerca del programma eseguibile ottimale e privo di errori è detta "debugging".

I programmi traduttori sono classificabili in 2 categorie principali:

Se la traduzione avviene **prima** delle esecuzione si parla di "**compilatori**", che richiedono un programma **compilatore** in grado di produrre il "**programma eseguibile**" partendo dal "**programma sorgente**". Per eseguire il programma eseguibile non è necessaria la presenza del compilatore, per cui l'eseguibile può essere trasferito anche su calcolatori che non sono dotati del compilatore.

Se la traduzione avviene **durante** l'esecuzione si parla di linguaggi **interattivi** che richiedono un programma "**interprete**" in grado di tradurre ed eseguire ogni singola istruzione del programma. In questo caso non abbiamo la distinzione tra programma sorgente e programma eseguibile, ma vi è un solo programma detto "**interpretabile**" (o "**script file**"). Il programma interpretabile può essere eseguito su altri calcolatori solo se sono dotati del relativo interprete.

I traduttori interattivi sono più semplici da utilizzare poiché non richiedono la fase di compilazione, ma sono ovviamente più lenti nell'esecuzione. Basic, Matlab e Perl sono principalmente interattivi, mentre C, C++ e Fortran sono prevalentemente dotati di compilatore; Java utilizza un meccanismo misto: richiede una compilazione in un linguaggio intermedio (Bytecode) che deve essere interpretato al momento dell'esecuzione (mediante la Java Virtual Machine).

## AGGIORNAMENTO E DISTRIBUZIONE

Qualunque tipo di software viene generalmente aggiornato in modo regolare per correggere eventuali errori di programmazione (**bug**) e per aggiungere nuove funzionalità (che a loro volta possono introdurre errori). Occorre per questo un sistema per distinguere tra loro le diverse versioni di uno stesso programma. La tecnica generalmente utilizzata è quella di aggiungere al nome del programma uno o più numeri separati da punto.

Il primo numero (**major release o versione**) viene incrementato in modo progressivo per ogni nuova versione che rende incompatibile il programma rispetto alle versioni precedenti o comunque introduce modifiche sostanziali.

Il secondo numero (**minor release o revisione**) viene eventualmente utilizzato per distinguere modifiche marginali, che mantengono compatibilità rispetto alla revisione precedente.

Un terzo eventuale numero può essere aggiunto per identificare nuove versioni di minor release in cui sono stati corretti errori di programmazione. La correzione degli errori di programmazione prende il nome di **debugging**.

Se il software viene scritto da più persone occorrono strumenti di lavoro collaborativo che gestiscano l'aggiornamento concorrente delle versioni. Lo strumento più utilizzato in ambiente Free Software è CVS – Concurrent Versions System

CVS consiste di un archivio in cui vengono immagazzinati i file che compongono il software in fase di sviluppo. Attraverso la rete un insieme di sviluppatori autorizzati possono aggiornare i singoli file del programma; CVS controlla ogni modifica informando gli altri sviluppatori e mantenendo traccia di tutte le modifiche apportate (per il ripristino di una versione precedente in caso di errori di programmazione).

Quando viene raggiunta una versione del software con i requisiti prefissati, il software viene estratto dall'archivio CVS e viene "impacchettato" in un formato adatto ad una agevole installazione.

## SISTEMI OPERATIVI

### INTRODUZIONE

Il sistema operativo è uno strato di software che opera direttamente sull'hardware, isolando i dettagli dell'architettura hardware e fornendo un insieme di funzionalità e di astrazioni ad alto livello.

Un sistema operativo è **mono-utente** se non è in grado di distinguere gli utenti che utilizzano il sistema.

Inoltre un sistema operativo è **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. Oggi i sistemi operativi sono multi-task e, la gran parte, multiutente. I sistemi operativi più utilizzati sono Microsoft Windows (nelle varie versioni: 98, NT, XP, 2000, ME, Vista) e Unix (con diversi "dialetti": Aix, Solaris, HPux, True64Unix, Linux).

Per facilitare e semplificare la costruzione del sistema operativo, vengono individuate diverse unità funzionali che lo compongono, le quali sono confinate in moduli indipendenti e sviluppate separatamente. Le unità funzionali principali sono:

- Gestione dei processi
- Gestione della memoria centrale

- Gestione delle periferiche
- File system

L'insieme di questi moduli costituisce il nucleo del sistema operativo, denominato anche Kernel. Il Sistema Operativo possiede inoltre una interfaccia per l'interazione con l'utente.

## UNIX, WINDOWS E LINUX

La storia di UNIX (nome derivato da un precedente progetto ambizioso ma fallimentare denominato Multics) inizia nei laboratori Bell della AT&T nel 1969, dove alcuni programmatori (Ken Thomson e altri) iniziarono a scrivere un sistema operativo per il PDP-7, uno dei primi calcolatori a transistor. Nel '73 Denis Richie (uno dei creatori del linguaggio C) collabora con Thomson per la riscrittura del Sistema Operativo in C, rendendolo quindi indipendente dall'hardware e facilmente trasportabile su altre piattaforme.

Unix inizia così a diffondersi nelle Università americane dove viene arricchito di nuove funzionalità. Nel 1975 l'Università di Berkeley crea una sua versione di Unix che chiama "BSD" che diventa il maggior concorrente dello Unix "System V" della AT&T. In seguito tutti i principali costruttori di calcolatori realizzano una loro versione di Unix per le loro WorkStations: IBM crea AIX, DIGITAL crea Ultrix (in seguito denominato Digital/Unix), Sun crea SunOS (in seguito Solaris), HP crea HPux, ecc.

Negli anni 80 Unix sembra destinato a diventare uno standard per tutti i calcolatori. In quegli anni però viene progettato il Personal Computer, basato su processore Intel 8086 e sistema operativo Microsoft DOS. Questo calcolatore possiede prestazioni e funzionalità molto limitate ma il costo molto basso ne favorisce una ampia diffusione. DOS era un sistema operativo Monotask, monoutente e con interfaccia utente a linea di comando.

Nel 1990 Microsoft sviluppa Windows 3.0 (e poi 3.1) che è una interfaccia utente di tipo grafico (GUI) per il sistema operativo DOS. Per poter gestire le finestre e il mouse dell'interfaccia è necessario un sistema operativo multitasking, per cui all'interno di Windows 3.1 viene inserito un multitasking rudimentale fatto a livello utente (di tipo cooperativo).

Nel 1995 Microsoft rilascia Windows95 che è un nuovo sistema operativo multitasking, ma ancora monoutente. In quel periodo Microsoft rilascia anche un sistema operativo multiutente, denominato NT, con caratteristiche più avanzate e destinato ad applicazioni di tipo server. Verso la metà degli anni 90 Unix sembra destinato ad una rapida estinzione a favore di Windows.

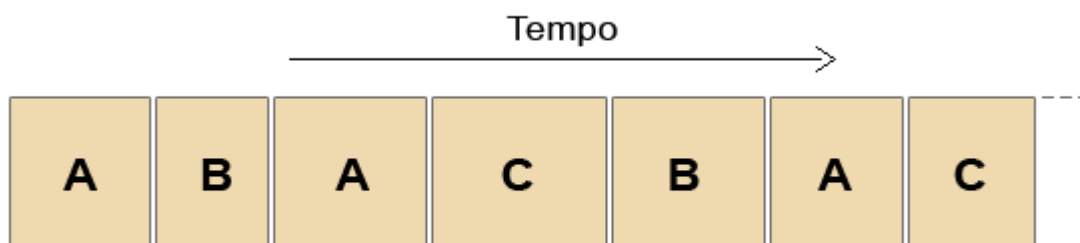
La riscossa di Unix arriva grazie a Linus Torvalds studente all'Università di Helsinki che nel 1991 ha creato Linux, un sistema operativo derivato da Unix, nato per sistemi basati su processori Intel x86. Linux è nato e si è sviluppato con la filosofia del software "Open" ed ha incentivato lo sviluppo di un ambiente completo di software aperto. Esistono attualmente diverse "distribuzioni" di Linux. Per distribuzione si intende il prodotto di qualcuno che raccoglie insieme vari programmi "open" componendo una suite completa (dal sistema operativo ai principali applicativi) che generalmente viene raccolta in uno o più CD in una forma di facile installazione. Questa "raccolta" di software è distribuita gratuitamente; chi compone la distribuzione basa i propri guadagni su servizi di assistenza

e aggiornamento, sui manuali di utilizzo ecc.,. Le principali distribuzioni sono Debian, Suse, Gentoo, Slackware, RedHat e derivati (Fedora, Ubuntu, Mandrake).

## LA GESTIONE DEI PROCESSI

Per processo (o task) si intende un programma in esecuzione.

Verso gli anni 60, con l'avvento delle prime periferiche ci si è resi conto che il processore aveva lunghe pause di attività in corrispondenza ad operazioni di Input/Output sulle periferiche (dischi, stampanti, ecc.) , che sono notoriamente più lente rispetto alla CPU. Per questo motivo sono stati ideati sistemi operativi "**multitasking**", in grado, cioè, di eseguire più task contemporaneamente. Quando un task (esempio task A) esegue una operazione di I/O la CPU interrompe l'esecuzione e passa al task successivo (esempio task B).



Successivamente, verso gli anni 70, con l'avvento dei terminali e quindi di un accesso interattivo con la macchina, i sistemi operativi si sono evoluti ulteriormente introducendo la funzionalità denominata "**time-sharing**".

Nei sistemi operativi Time-Sharing un task se non incontra operazioni di Input/Output entro un certo tempo (dell'ordine di decine di millisecondi) viene egualmente sospeso dall'esecuzione. In questo modo il tempo della CPU non può essere "monopolizzato" da un processo ma ruota rapidamente tra tutti i processi in attesa.

I sistemi operativi Time-Sharing sono spesso anche "**MultiUtente**" nel senso che devono gestire la presenza di diversi utenti sulla macchina, dove ogni utente deve essere identificato per potergli attribuire in modo personalizzato una politica di accesso alle risorse del calcolatore. In questa rincorsa ad aggiungere nuove funzionalità, il sistema operativo è divenuto un insieme molto complesso di programmi e dati. 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 (o, in alcuni processori, dell'istruzione in corso di esecuzione).

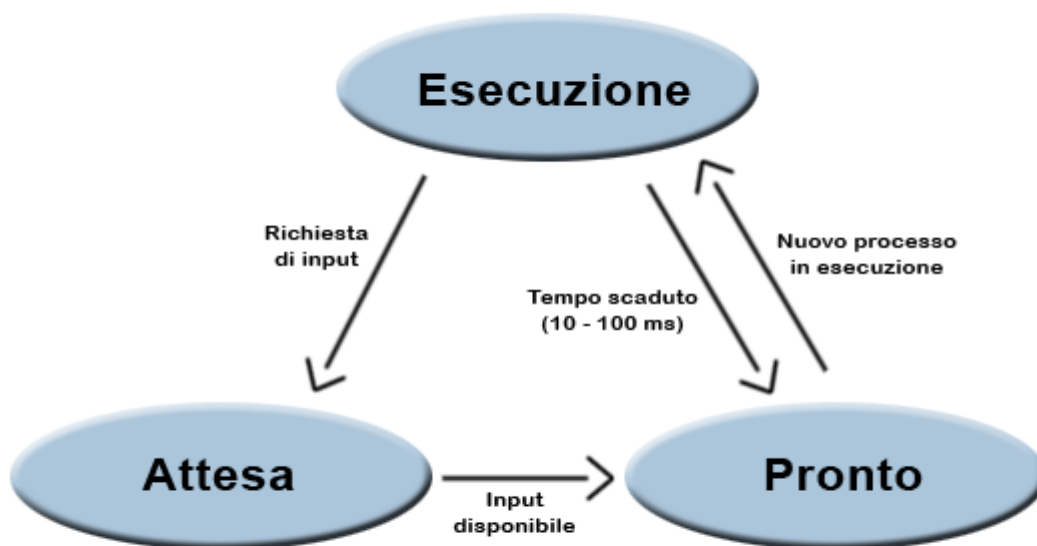
Il sistema operativo dovrà gestire anche lo stato del processo relativamente alla competizione per l'accesso al processore. Per questo esistono 3 stati possibili in cui può trovarsi un processo:

-E: **esecuzione** (running)

-P: **pronto** (ready)

-A: **attesa** (blocked)

Quando il processo in esecuzione incontra una operazione di I/O (esempio: attesa di dati dalla tastiera, dal disco, dalla rete, ecc) viene spostato dallo stato di "esecuzione" allo stato di "attesa"; al termine dell'operazione di I/O lo stato del processo commuta da "attesa" a "pronto". Il processo in esecuzione esce comunque dallo stato di esecuzione se supera il tempo a disposizione. In questo caso lo stato del processo viene commutato direttamente da "esecuzione" a "pronto".



Il sistema operativo avrà il compito di determinare e applicare il migliore metodo possibile per la scelta del processo da mettere in esecuzione tra quelli nello stato di "pronto", tenendo conto delle diverse priorità che i processi possono avere.

## LA GESTIONE DELLA MEMORIA

Ogni processo possiede un proprio spazio di memoria in cui risiedono istruzioni e dati del programma, inoltre tutti i processi condividono uno spazio di memoria (comune) in cui risiedono istruzioni e dati del sistema operativo.

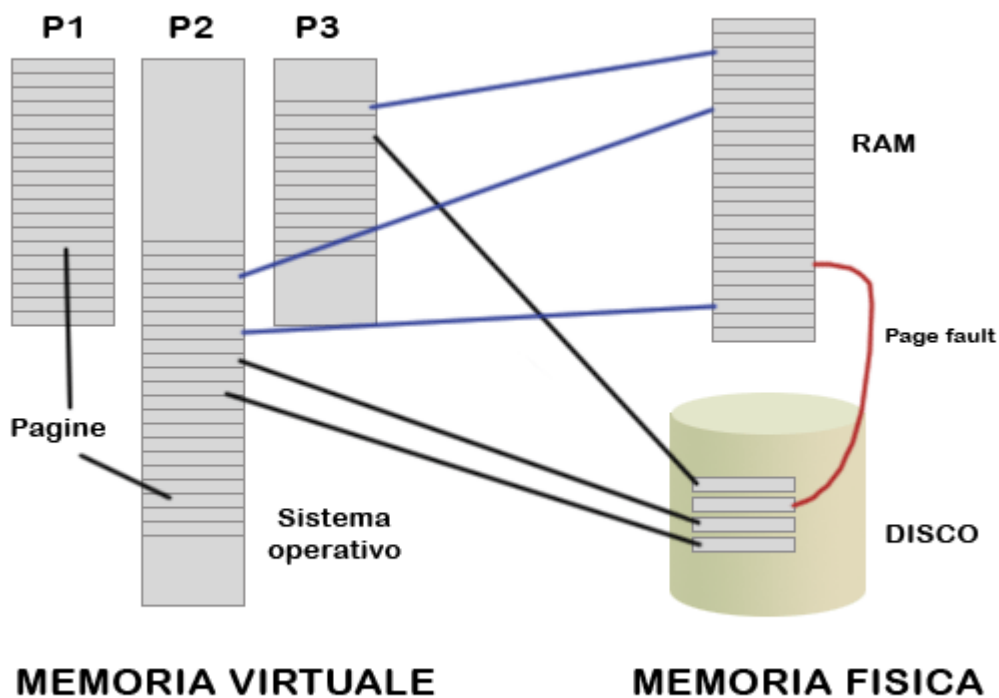


La dimensione dello spazio di memoria propria di ogni processo dipende dalla dimensione del programma e dei dati che utilizzerà. Questi spazi di memoria (tutti i processi più il sistema operativo) devono condividere la memoria fisica (RAM) del calcolatore.

La tecnica attualmente utilizzata dai più diffusi sistemi operativi (Unix/Linux e Windows) per gestire al meglio la memoria del calcolatore prende il nome di paginazione. Mediante la paginazione in sistema operativo fornisce ad ogni processo una area di memoria "virtuale". Per macchine a 32 bit quest'area è di  $2^{32}$  byte, ovvero 4GByte, di cui generalmente 2Gbyte sono a disposizione del processo per programmi e dati, mentre 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. Questa associazione è ottenuta suddividendo lo spazio virtuale di ogni processo in "Pagine Virtuali" di dimensione fissa (tipicamente 4 KByte); nella memoria centrale (RAM) e in parte della memoria secondaria (una porzione del disco fisso) vengono ricavate "pagine fisiche" della stessa dimensione.

Mediante opportune tabelle (una per ogni processo) le pagine virtuali effettivamente utilizzate vengono "mappate" dinamicamente sulle pagine fisiche della RAM o del Disco a seconda del loro utilizzo. Infatti le pagine virtuali possono essere accedute dal processo unicamente se sono mappate sulla memoria centrale (RAM), per cui 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, occorrerà un intervento del sistema operativo per la migrazione della pagina in oggetto da Disco a RAM. Questa procedura è denominata "Page-fault".



La paginazione è trasparente per i processi. La paginazione può però incidere sui tempi di esecuzione dei processi poiché se l'elaboratore possiede una quantità di memoria RAM

insufficiente rispetto alle richieste dei processi, il sistema operativo sarà costretto a continui pagefault (accessi a disco) con conseguenti ritardi nei tempi di esecuzione.

Nei sistemi operativi Windows NT/2000 lo spazio su disco per la paginazione è nel file C:\pagefile.sys che può essere visualizzato dalla shell DOS con il comando **DIR C:\ /A**

La gestione di questo file (dimensione iniziale e dimensione massima) è configurabile seguendo il percorso:

Pannello di controllo -> Sistema -> Avanzate -> Opzioni Prestazioni -> Memoria Virtuale

## MEMORIA SECONDARIA, FILE E FILE-SYSTEM

Ogni processo in esecuzione sul calcolatore esegue elaborazioni su dati prelevati da dispositivi di Input e deposita i risultati su dispositivi di Output.

La tastiera è il dispositivo di Input standard, mentre il video è il dispositivo standard per l'Output.

Un'altro importante strumento per l'Input/Output è il File che è una astrazione fornita dal sistema operativo per accedere ai dispositivi della memoria secondaria.

### LA MEMORIA SECONDARIA

È chiamata anche memoria di massa e racchiude tutti quei dispositivi che consentono all'utente la memorizzazione in modo permanente. Ha quindi la funzione di archivio per i programmi e i dati del calcolatore. È realizzata prevalentemente con Hard Disk, ma anche con Floppy, CD-Rom, DVD, Flash Memory e nastri magnetici.

Per semplificare e razionalizzare l'accesso alle informazioni contenute in questi dispositivi il sistema operativo nasconde le strutture dati di basso livello mostrando all'utente una struttura dati gerarchica basata su **“File”** e **“Directory”**.

### I FILE:

Un file è un insieme di informazioni correlate e registrate nella memoria secondaria. In generale un file è formato da una sequenza di byte il cui significato è definito dal creatore e dall'utente del file stesso.

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 ultimo accesso al file.**
- **Proprietà e protezione:** Informazioni per il controllo degli accessi al file.

Il nome del file generalmente e` composto da 2 parti separate da punto. (Esempio MIOFILE.TXT)

La seconda parte prende il nome di estensione. L'estensione viene utilizzata convenzionalmente per identificare il tipo di file in base al nome. Le estensioni possono essere utilizzate per associare automaticamente ogni tipologia di file dati al programma che lo gestisce. Questo meccanismo e` utilizzato nella Posta elettronica, nei browser WWW, nelle interfacce GUI, ecc.

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.

Le principali **operazioni** sui file che vengono generalmente fornite dal sistema operativo sono: Creazione, apertura, chiusura, scrittura, lettura, riposizionamento, cancellazione. Lettura e/o modifica degli attributi.

---

#### LE DIRECTORY:

La directory è un'altra astrazione che il sistema operativo fornisce all'utente per la gestione ordinata dei file. Tramite le directory i file possono essere memorizzati in una struttura organizzata che generalmente e` un albero ad altezza arbitraria. Ogni file viene collocato in una particolare directory dell'albero ed e` identificato univocamente dal nome del file e dal percorso per raggiungere la directory che lo contiene a partire dalla radice dell'albero (root directory).

Nei file-system Unix:

`/percorso/directory/nomefile.estensione`

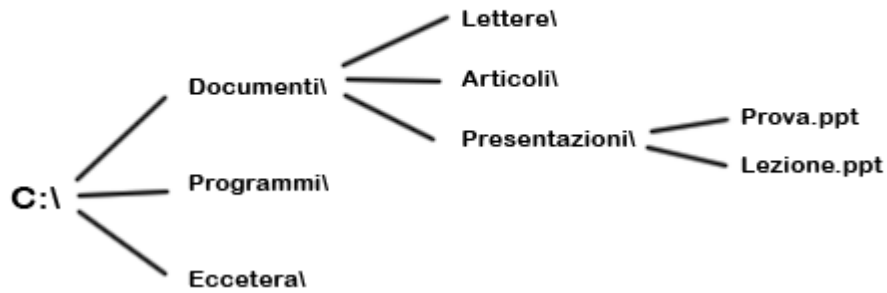
Esempio: `/etc/mime.types`

Nei file system di Windows:

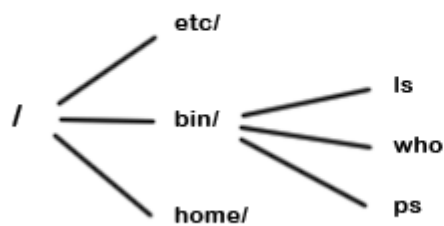
`dispositivo:\percorso\directory\nomefile.estensione`

Esempio: `C:\windows\notepad.exe`

## Windows



## Linux



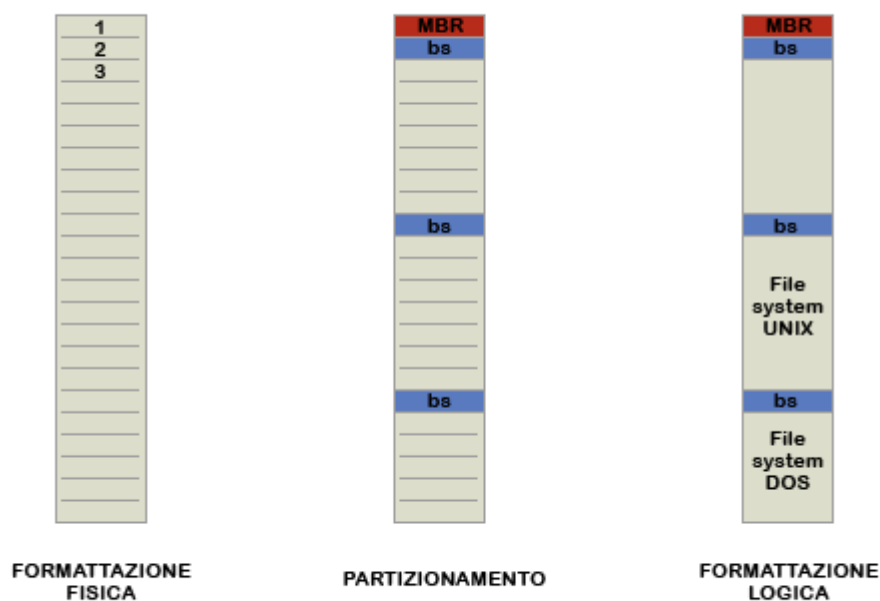
---

### PARTIZIONAMENTO DI UN DISCO:

Abbiamo visto che un Disco fisso è logicamente suddiviso in una sequenza di blocchi di dimensione fissa (generalmente 512 byte oppure 1KB). Un disco fisso può essere suddiviso in diversi dischi logici denominate "Partizioni".

Il primo blocco del disco fisso è denominato Master Boot Record (MBR), mentre il primo blocco di ogni partizione è denominato "Boot Sector". MBR e Boot Sector contengono informazioni riguardo l'avvio del sistema operativo eventualmente contenuto nel disco fisso.

Infatti al momento dell'accensione del computer il programma residente nella ROM dopo avere testato l'hardware presente sul computer va alla ricerca di un programma per il caricamento del sistema operativo. Questo programma viene cercato generalmente nel MBR del disco fisso. Il MBR può reindirizzare questa ricerca al Boot Sector di una delle Partizioni.



#### IL FILE-SYSTEM:

Il sistema operativo, mediante il File-System, implementa file e directory sulla memoria secondaria, utilizzando una piccola parte dello spazio su disco. L'installazione del File-System su di una partizione del disco prende anche il nome di formattazione logica.

Il File-System ha il compito di nascondere l'organizzazione in blocchi, mostrando all'utente i dati organizzati in file e directory. Per questo deve gestire i blocchi del disco, assegnando i blocchi liberi a nuovi file richiesti dall'utente. Se lo spazio libero è suddiviso in numerosi piccoli spazi i nuovi file risulteranno frammentati, ovvero composti da blocchi sparsi in punti diversi del disco. Questa frammentazione degrada le prestazioni del disco; per questo esistono programmi "deframmentatori" che hanno il compito di ricompattare i blocchi dei file e gli spazi liberi.

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".

**NTFS:** 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.

## 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).

Le funzionalità 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 e configurazione dei driver
  - Gestione dei dischi (formattazione, installazione file system)
  - Gestione delle 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.

---

### 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 è semplificata dall'utilizzo di icone attivabili (mediante il sistema di puntamento) le quali ricordano graficamente la funzione associata. Questa modalità è 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.