Sistemi operativi

Il software

- Il software può essere diviso un due grandi classi:
  - i ***programmi di sistema*** che gestiscono le operazioni del sistema di elaborazione;
  - i ***programmi applicativi*** che risolvono i problemi dei loro utilizzatori.
- L'insieme dei *Programmi di Sistema* viene comunemente identificato con il nome di **Sistema Operativo (SO).**

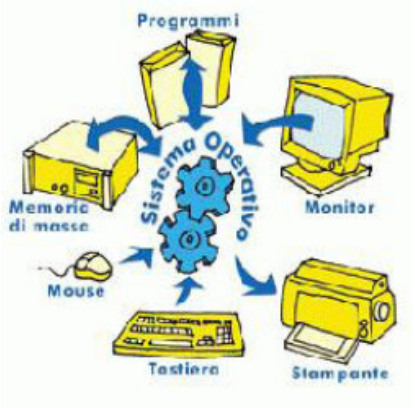
Il **sistema operativo** (abbreviato in **SO**, o all'inglese **OS**, *operating system*) è il programma responsabile del diretto controllo e gestione dell'hardware che costituisce un computer e delle operazioni di base.
Si occupa dei processi che vengono eseguiti e della gestione degli accessi degli utenti. Compito del sistema operativo è inoltre quello di fare da interfaccia tra le risorse hardware e i programmi applicativi.
Il SO può essere inteso come uno strumento che virtualizza le caratteristiche dell'hardware, offrendo all'utente la visione di una macchina astratta più potente e più semplice da utilizzare di quella fisicamente disponibile.

Alcuni dei programmi del sistema operativo interagiscono con le componenti hardware che costituiscono i calcolatori, quindi questi devono conoscere le caratteristiche dei dispositivi in modo da poterli controllare e adoperare. Inoltre il sistema operativo deve essere in grado di accedere anche a dispositivi non previsti inizialmente, purché questi vengano forniti con il corredo del software necessario, comunemente chiamato **device driver** (*pilota del dispositivo*). In altri termini, lo stesso sistema operativo deve poter essere ampliato e personalizzato a seconda della macchina su cui viene installato. Sui sistemi operativi moderni questo compito viene svolto spesso in modo automatico, o comunque senza intervento di personale specializzato. I dispositivi realizzati in tecnologia **plug and play** (inserisci e usa) colloquiano con il sistema in modo tale da "farsi riconoscere" e richiedere l'installazione dei driver corretti. Il sistema operativo è così in grado di configurare tali dispositivi in modo che non si creino conflitti con dispositivi preesistenti e che l'insieme funzioni correttamente.
Tramite il sistema operativo, dunque, gli utenti possono utilizzare le varie componenti hardware del sistema. Infatti è possibile caricare in memoria centrale un programma da eseguire, attivare la CPU, richiedere la lettura di un dato memorizzato su memoria di massa, effettuare delle stampe. E' il sistema operativo che si preoccupa di effettuare queste operazioni, frapponendosi tra la richiesta dell'utente e l'hardware della macchina. Un'altra funzione del sistema operativo è quella di ripartire una risorsa che è fisicamente limitata (una sola CPU, una memoria centrale che ha una dimensione ben definita) fra le tante richieste possibili. Queste funzioni sono espletate tramite le ***politiche di gestione*** che definiscono le regole per una gestione razionale delle risorse.

Riassumendo

Il **sistema operativo**:

- è uno *strato software* che *opera direttamente sull'hardware;*
- rende totalmente disponibile all'utente l'hardware del calcolatore senza che l'utente debba conoscere le caratteristiche tecniche dell'hardware (isola gli utenti dall'architettura sottostante) e fornisce un insieme di funzionalità di alto livello;
- permette lo svolgimento di operazioni quali la copia di un file o l'esecuzione di un programma;
- opera le azioni necessarie a caricare i programmi in memoria centrale, eseguirli, leggere e/o scrivere dati da/su memoria di massa e periferiche.



Scopo del Sistema Operativo

- Gestione ***efficiente*** delle risorse del sistema di elaborazione.
- Creazione di un'***interfaccia amichevole*** (semplice e intuitiva) tra l'uomo e la macchina.
- Nascondere i dettagli tecnici della macchina.

Attività svolte dal Sistema Operativo

- Gestione della memoria di massa (file system);
- Gestione della memoria RAM;
- Efficienza nell'uso delle risorse (processori, memoria, dischi, etc.)
- Gestione e coordinamento dei processi;
- Gestione dell' interfaccia utente;
- Accesso simultaneo di più utenti alla stessa macchina;
- Esecuzione simultaneamente di più processi sulla stessa macchina.

Caricamento del Sistema Operativo

In un sistema di elaborazione l'hardware da solo non serve a nulla. Ha bisogno di software cioè di programmi che consentano il colloquio utente-macchina.

Quando si accende un computer, il primo software che viene messo in esecuzione è **una parte del SO** che si chiama ***bootstrap***, e serve:

- a eseguire il ***POST*** (Power On Self Test). Si tratta di una serie di test diagnostici per verificare il corretto funzionamento dell'hardware della scheda madre (conteggio della dimensione della RAM, della cache,..) e controlla la presenza e il corretto funzionamento di tutti i dispositivi periferici (tastiera, mouse, lettore CD e DVD, …).
- a rendere utilizzabili i dischi fissi (nei quali potrà cercare il resto del sistema operativo)
- a trasferire il resto del SO dal disco fisso alla memoria RAM (dalla quale potrà essere eseguito).

Nei primi personal computer il SO era piccolissimo e stava tutto in ROM. Quando accendevi, era subito disponibile dopo pochissimi secondi. Nei PC attuali, occorre attendere che il SO sia in RAM per poterlo eseguire.

Una volta in RAM, il SO può essere eseguito dalla CPU. Alla fine della fase di bootstrap, cioè dell'accensione, il computer è in attesa dell'intervento da parte dell'utente.

Le tipologie dei sistemi operativi

I sistemi operativi trovarono la loro origine verso la fine degli anni Quaranta, quando fu sentita in modo pressante la necessità di avere a disposizione non solo un'interfaccia "amichevole" che sollevasse il programmatore dalla gestione diretta dei dispositivi di ingresso/uscita, ma anche supporti che velocizzassero alcune operazioni di base (caricamento, assemblaggio, messa a punto, ecc.) a fronte dello sviluppo delle prestazioni hardware dei calcolatori. Da allora i sistemi operativi si sono andati via via evolvendo, passando dalle strutture più semplici dei sistemi dedicati a quelle odierne che sovraintendono addirittura al colloquio tra più computer posti in rete telematica tra loro. Di seguito sono presentate le fasi più salienti di questa evoluzione, fornendo per ciascun tipo di sistema operativo una breve descrizione delle sue caratteristiche più significative.

Sistemi dedicati

Un calcolatore, nella sua struttura iniziale, era in grado di gestire **un solo programma per volta (monoprogrammazione)** che non solo doveva essere completamente residente in memoria, ma a cui venivano destinate tutte le risorse a disposizione (processore, memoria RAM, periferiche….).

Si parla, in questo caso, di *sistema dedicato e* in esso il SO, peraltro molto semplice, aveva la funzione principale di supportare, oltre alle operazioni di assemblaggio, caricamento, inizializzazione e terminazione di programmi, anche la gestione di semplici procedure di ingresso e uscita (per esempio, quelle per la memorizzazione in successive locazioni di memoria dei codici binari letti da un pacco di schede perforate o la stampa dei risultati di una elaborazione su un tabulato).

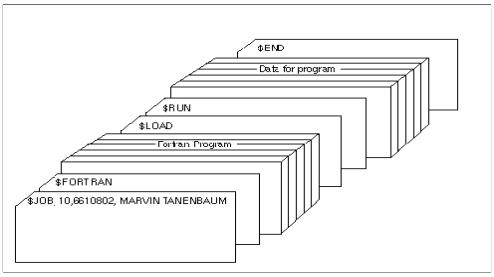
Elaborazione a lotti o batch

Verso la fine degli anni Cinquanta, al crescere della velocità delle unità di elaborazione, risultò inaccettabile che un sistema (il cui costo era estremamente elevato) fosse a disposizione di un unico utilizzatore che occupava tempo-macchina con lunghe e onerose procedure di caricamento e gestione di programmi, e quindi fu adottata una nuova organizzazione del lavoro, detta ***elaborazione a lotti*** o ***batch***. Nell'elaborazione a lotti il SO raccoglie i **lavori** *(job)* presentati da diversi utenti e li esegue in rapida successione, eliminando i tempi morti provocati dal caricamento manuale. Nel caso l'esecuzione di un passo di un job produca un errore, l'esecuzione di quel lavoro viene bloccata e si passa al successivo.

Con questo tipo di organizzazione i lavori degli utenti devono essere "autonomi", cioè devono contenere tutti i comandi necessari per attivare i singoli passi *(job step)* che consentono di eseguire e portare a compimento un programma senza alcun intervento di un operatore umano. Si utilizza a questo scopo un preciso linguaggio di controllo *(Job Control Language)* che permette di descrivere i passi che il lavoro deve percorrere.

Il singolo programmatore, in questo caso, deve (**fasi del lavoro batch**):

1. sviluppare il proprio programma;
2. costruire una sequenza di comandi, espressa nel linguaggio di controllo, che specifica le fasi da far attraversare al proprio programma (per esempio, compilazione, collegamento ed esecuzione);
3. caricare il tutto su un opportuno supporto di memoria (es. schede perforate o nastro magnetico);
4. consegnare il lavoro all'operatore e tornare successivamente per il ritiro dei risultati.



I sistemi batch garantiscono sì la sequenzialità dei lavori, ma presentano lo svantaggio di far perdere all'utilizzatore "il contatto" con l'elaboratore, infatti anche piccole modifiche da apportare al programma costringono il programmatore a ripercorrere l'intera procedura dall'inizio.
Inoltre la **CPU viene comunque sottoutilizzata** perché, durante le operazioni di I/O, deve adeguarsi alla bassa velocità delle periferiche quali i lettori di schede, i nastri, i dischi e le stampanti.

Per questo motivo, soprattutto con la diffusione di unità periferiche veloci e di grande capacità (dischi e nastri), i sistemi operativi di tipo batch vennero sostituiti con altri che meglio potevano sfruttare la potenzialità delle nuove risorse.

Sistemi multiprogrammati

Ben presto si arrivò alla conclusione che per migliorare la situazione occorreva soprattutto diminuire i tempi legati alla gestione delle periferiche esterne (unità di input e output che operavano ad una velocità di molte volte minore alla CPU). La soluzione più ovvia fu quella di organizzare il lavoro del sistema in modo che **più programmi potessero *risiedere contemporaneamente in memoria (RAM)*** in aree distinte.

Si ottenne così un sistema **multiprogrammato,** caratterizzato dalla presenza simultanea di **diversi programmi nella memoria centrale** e dalla loro esecuzione alternata.

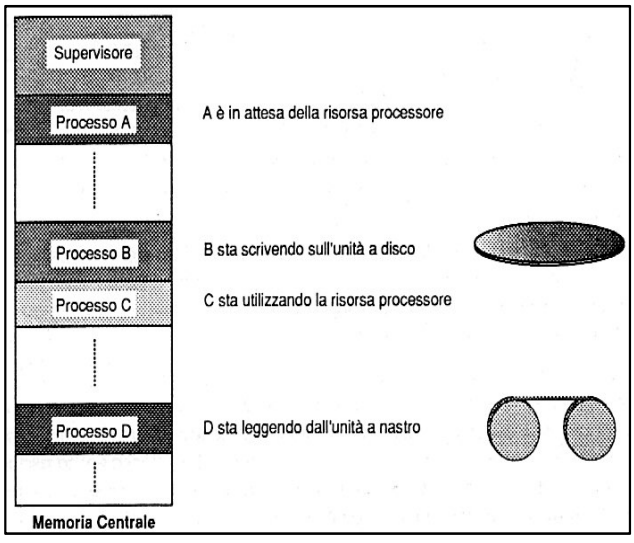
Il SO era detto a **partizioni di tempo** (**time sharing**). Esso assegnava il processore ad un altro programma, ogni volta che quello che in quel momento era in esecuzione doveva fermarsi in attesa della terminazione di una operazione di I/O (disco, stampante, tastiera ...).

In questo modo il processore, invece di attendere inattivo la fine dell'operazione, veniva assegnato a un altro lavoro (job). L'esecuzione del lavoro sospeso, una volta completata l'operazione di I/O, poteva riprendere in un secondo momento quando il lavoro successivo era terminato o se richiedeva anch'esso una operazione di I/O.

Dall'esterno si aveva l'impressione che più programmi venissero eseguiti contemporaneamente (**parallelismo**): in realtà, in presenza di una sola CPU, un solo programma utente può essere eseguito in un certo istante.

In questo tipo di sistemi multiprogrammati si presentarono però nuovi problemi ed esigenze, in particolare la necessità di risolvere:

- i ***problemi di protezione****:* come proteggere i programmi degli utenti da intrusioni di altri programmi (ad esempio che due aree dati nella memoria RAM non venissero a sovrapporsi anche solo parzialmente a seguito della esecuzione di due programmi diversi) e come salvaguardare il sistema di elaborazione dai suoi utilizzatori;
- i ***problemi di condivisione****:* delle risorse tra i diversi programmi in memoria (per esempio due programmi dei quali si richiede l'esecuzione e che quindi necessitano entrambi della CPU) .



Successivamente, con il diminuire dei costi dell'hardware, si pensò di "aiutare" il processore centrale, utilizzando processori dedicati (DMA) per le operazioni di I/O e, ancora meglio, di costruire calcolatori multiprocessore.

Ovviamente, con l'aumentare della complessità del sistema, cresceva anche il tempo di CPU necessario al SO e di conseguenza l'**overhead di sistema** (il sovraccarico del sistema). Non dobbiamo dimenticare che il SO è costituito da molti programmi che devono a loro volta essere eseguiti dalla CPU per coordinare l'esecuzione dei programmi degli utenti. Inoltre, dato che la dimensione del SO non permetteva più di mantenerlo sempre completamente in memoria RAM, lo stesso fu diviso in una parte che rimaneva sempre in memoria, detta ***supervisore***, e in diverse routine memorizzate su disco e richiamate solo all'occorrenza.
Il **supervisore** quindi in tale organizzazione rappresenta il nucleo del sistema operativo preposto alla gestione dei processi e dei processori.

Sistemi in tempo reale o real-time

Si usano per il controllo dei processi industriali, ed in generale nei casi in cui sia necessario che il SO intervenga tempestivamente.
Questi sistemi operativi sono al servizio di una specifica applicazione che ha dei vincoli precisi nei **tempi di risposta**, deve quindi *garantire un tempo massimo* entro il quale mandare in esecuzione un programma a seguito di una richiesta (gestione allarme, sistema airbag di un'auto, sistema antibloccaggio ruote ABS, controllo di un aereo teleguidato, controllo di processi…)
Di solito NON interagiscono con l'utente, ma piuttosto con l'ambiente (sensori, trasduttori) per mantenere il sistema in funzione correttamente. In generale si ha un sistema real-time quando il tempo che passa dalla richiesta di esecuzione di un processo al completamento della stessa è minore del tempo fissato.
Talvolta si usa dire "in tempo reale" riferendosi a sistemi nei quali le cose "accadono subito", per esempio per la "prenotazione in tempo reale" di un volo aereo, nata dall'abuso del gergo informatico da parte di profani.
Per questi sistemi è stata coniata l'espressione "*real time in senso lato*" per differenziarli da quelli che sono classificati come veri "real time".

Sistemi Operativi di Rete

I SO di rete sono un'estensione dei S O standard con, in più, i servizi per la gestione di risorse in rete locale
Le risorse gestite sono: **uno o più server di rete**, più **stampanti di rete, una o più reti fisiche,** un **numero potenzialmente grande di utenti.**

Le funzioni sono molteplici e più complesse di quelle di un SO standard:

- gestire un file system per dati, applicazioni, profili utente e periferiche di rete
- coordinare tutte le risorse e i servizi disponibili
- elaborare le richieste degli utenti e richiedere agli utenti le informazioni per l'accesso alla rete, convalidare gli account, applicare le limitazioni
- gestire gli utenti connessi in modalità locale e remota e supportare la protezione del sistema
- gestire l'interconnessione tra reti locali
- supportare le funzioni client/server

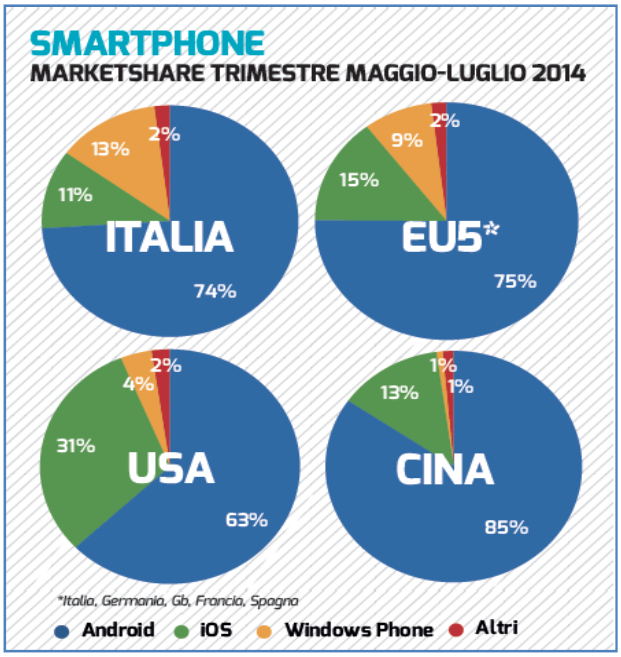
Sistemi per la multimedialità

Fino a qualche anno fa i sistemi operativi dovevano gestire le tipiche applicazioni che utilizzavano dati tradizionali, applicazioni come fogli elettronici, editor di testo, ecc.; negli ultimi anni hanno acquistato sempre più importanza applicazioni che utilizzano dati multimediali, come file video e audio, che necessitano di particolarità che si riflettono sulla progettazione dei sistemi operativi che devono fornire supporto anche a tali applicazioni. A tale proposito è facilmente comprensibile come le problematiche relative alla visione in diretta, tramite il web, degli eventi sportivi delle olimpiadi di Pechino in 6 finestre aperte contemporaneamente in un browser (programma di navigazione in Internet) sono differenti rispetto a quelle relative alla gestione di un semplice documento predisposto con un programma per la videoscrittura.

Sistemi per dispositivi mobili

Con il termine **dispositivi mobili** (in inglese *mobile device*) si intendono tutti quei dispositivi elettronici che sono pienamente utilizzabili seguendo la mobilità dell'utente quali telefoni cellulari, palmari, smartphone, tablet, laptop, ricevitori GPS ecc. Hanno dimensioni e peso ridotti tali da poter essere trasportati dall'utente. Storicamente i primi dispositivi di questo tipo sono stati i telefoni cellulari di prima generazione.
Questi dispositivi sono sempre più simili nelle prestazioni ai dispositivi fissi pur mantenendo le loro peculiarità in termini di dimensioni ridotte. La loro evoluzione sembra non arrestarsi.

I primi **smartphone** combinavano le funzioni di un computer palmare con quelle di un telefono mobile. I modelli più recenti si sono arricchiti della funzionalità di dispositivi multimediali in grado di riprodurre musica, scattare foto e girare video. Sono dotati di touchscreen ad alta risoluzione e web browser che sono in grado di caricare sia normali pagine web sia siti web appositamente creati per i dispositivi mobili consentendo una navigazione web completa.

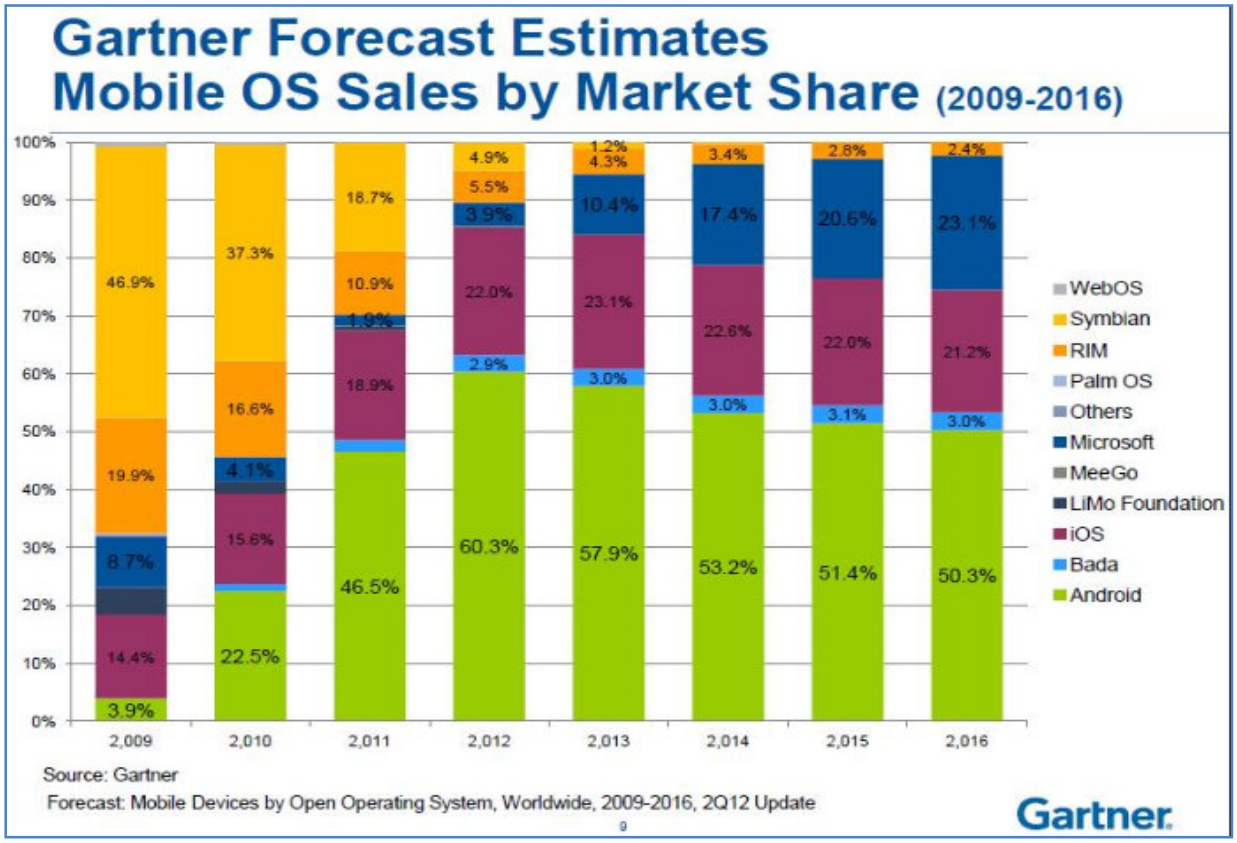


Caratteristica diffusa è inoltre quella di poter installare funzionalità aggiuntive attraverso le cosiddette **App** (applicazioni dedicate ai dispositivi mobili).

Oggi i dispositivi della quarta generazione dei servizi di telefonia mobile supportano la veloce tecnologia 4G per inviare e ricevere informazioni digitali.

I sistemi operativi che gestiscono tali dispositivi devono avere delle caratteristiche particolari basate, ad esempio, sull'uso di memoria virtuale e sul risparmio energetico, condizionate dall'utilizzo di CPU più piccole del normale ma anche più lente, da metodi di inserimento dati limitati (piccole tastiere anche simulate sullo schermo, touch screen, ecc.), da schermi per la visualizzazione delle informazioni di piccole dimensioni, ecc.

Il mercato dei dispositivi mobili, e degli smartphone in particolare, è ad oggi territorio di conquista di tre sistemi operativi: **Android** di Google, **iOS** di Apple e **Windows Phon**e (WP) di Microsoft (immagine riferita al 2014). Ad oggi Google, Apple e Microsoft sono i vincitori evidenti nell'ecosistema dei dispositivi mobili, ma il mercato mondiale è soggetto a mutazioni forti in periodi brevi, anche in pochi anni, come mostrano le proiezioni di Gartner relative agli anni 2009-2016.



Classificazione dei SO

- **In base alle modalità di gestione dei programmi** abbiamo visto che si può avere:
  - ***Monoprogrammazione*** (un solo programma alla volta in memoria)
  - ***Multiprogrammazione*** (più programmi presenti in memoria, apparentemente eseguiti contemporaneamente, in realtà in *time-sharing*, suddividendo il tempo di esecuzione in intervalli molto piccoli e assegnando a turno le risorse ai diversi programmi)
- **In base al tipo di accesso fornito agli utenti:**
  - ***SO monoutente*** (un solo utente può usare la macchina) è tipico nei PC; l'intero sistema hw/sw è ***dedicato*** ad un singolo utente.

    Sistemi dedicati: sono i sistemi che prevedono l'utilizzo da parte di un solo utente per volta. In questa categoria rientrano i personal computer e le workstation. I sistemi dedicati attuali supportano il **multitasking** che è un tipo particolare di multiprogrammazione nella quale più applicazioni (**task**), richieste da un singolo utente, vengono eseguite contemporaneamente sulla stessa macchina.
  - ***SO multiutente*** (più utenti contemporaneamente possono interagire con la stessa macchina). In questo caso diversi utenti condividono lo stesso sistema hw/sw; il SO nasconde a ciascun utente la presenza degli altri, dando l'impressione che il sistema (unità di elaborazione, memoria, periferiche, etc.) gli sia interamente dedicato.

    Nei sistemi multiutente, all'aumentare del carico complessivo del sistema, ovvero, al crescere del numero delle richieste di elaborazione, il sistema può diventare sovraccarico e fornire prestazioni percepibilmente scadenti.

Struttura del Sistema Operativo

L'obiettivo del SO consiste nell'ottimizzare le prestazioni del sistema informatico, determinando le politiche migliori di gestione delle risorse sotto il suo controllo.
Il SO è un insieme di programmi molto complesso ed articolato, soprattutto in contesto multi–utente.
Per facilitarne il progetto, ed isolarne le varie componenti, il SO è organizzato per **strati funzionali o moduli**, con una struttura cosiddetta "a cipolla".
Ogni strato è dedicato a svolgere una determinata funzione, realizzando una macchina virtuale; i vari strati interagiscono poi tra loro secondo regole precise al fine di realizzare le funzionalità di base del sistema di elaborazione. La macchina fisica (**hardware**) + il **sistema operativo** costituiscono la **macchina estesa o macchina astratta**.

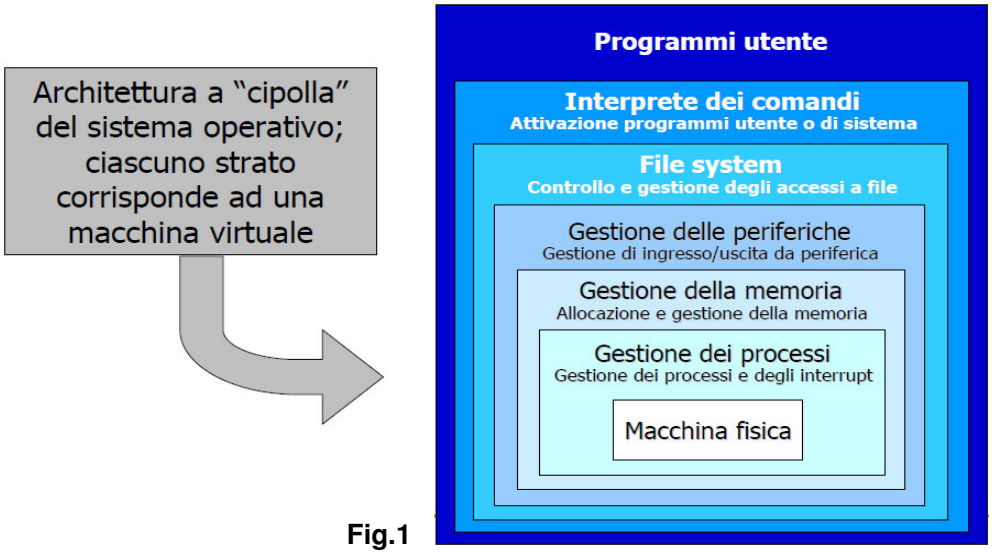


Fig.1

Il **kernel** o **nucleo** è il *cuore* di un sistema operativo. Esso ha il compito di fornire ai moduli che compongono il sistema operativo e ai programmi in esecuzione sul computer le funzioni fondamentali ed un accesso controllato all'hardware, sollevandoli dai dettagli della sua gestione. Si occupa dell'esecuzione dei programmi e della risposta agli eventi esterni generati dalle periferiche. Scopo principale è quello di gestire i **processi** corrispondenti ai programmi che sono contemporaneamente attivi, attuando una politica di alternanza (scheduling) nell'accesso alla CPU da parte dei processi in esecuzione.

Il **nucleo** può comprendere solo lo **strato più interno** (dedicato alla *gestione dei processi)* oppure **i primi tre strati più interni** (dedicati alla *gestione dei processi, della memoria* e *delle periferiche*) a seconda che l'organizzazione del SO segua il **modello monolitico** (Fig.2) o il **modello a microkernel** (Fig.3).

Modalità di funzionamento del processore

- **Stato utente**
  Modalità di funzionamento dell'hardware e in particolare del processore (CPU), che permette **l'accesso solo a un sottoinsieme delle risorse** disponibili (es.: non si può accedere alle istruzioni che istruiscono le interfacce di I/O, si può accedere solo ad una parte della memoria RAM etc.). Questo è lo stato tipico in cui il processore esegue i **programmi degli utenti** che per poter effettuare certe operazioni privilegiate devono ricorrere all'intervento del SO con una interruzione di **supervisor call** (per esempio la richiesta di stampa di un documento o il salvataggio su una memoria di massa, l'acquisizione di un dato da tastiera ecc.,).
  I programmi che girano in stato utente:
  - *richiedono*, quando ne hanno necessità, dei servizi al SO tramite invocazione di sottoprogrammi speciali (*chiamate di sistema)*.
  - Il sistema operativo decide come e quando effettuare il servizio
  - Il sistema operativo può interrompere un programma che gira in stato utente per eseguire altri programmi o per effettuare operazioni di 'gestione' della macchina

- **Stato supervisore** o **kernel**
  Modalità privilegiata che permette **l'accesso a tutte le risorse**. Questo è lo stato tipico in cui il processore esegue i **programmi del nucleo del SO**.

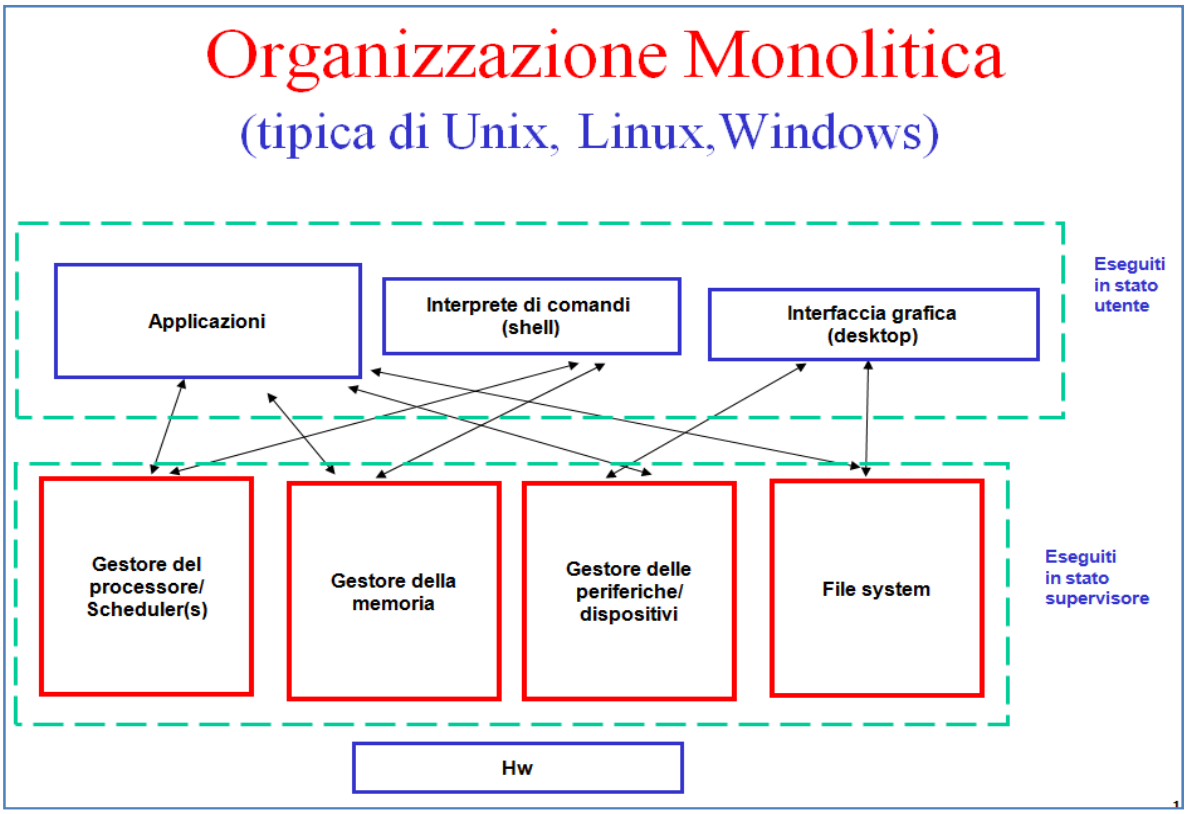

Organizzazione Monolitica
(tipica di Unix, Linux,Windows)
Applicazioni
Interprete di comandi (shell)
Interfaccia grafica (desktop)
Eseguiti in stato utente
Gestore del processore/ Scheduler(s)
Gestore della memoria
Gestore delle periferiche/ dispositivi
File system
Eseguiti in stato supervisore
Hw

Fig.2

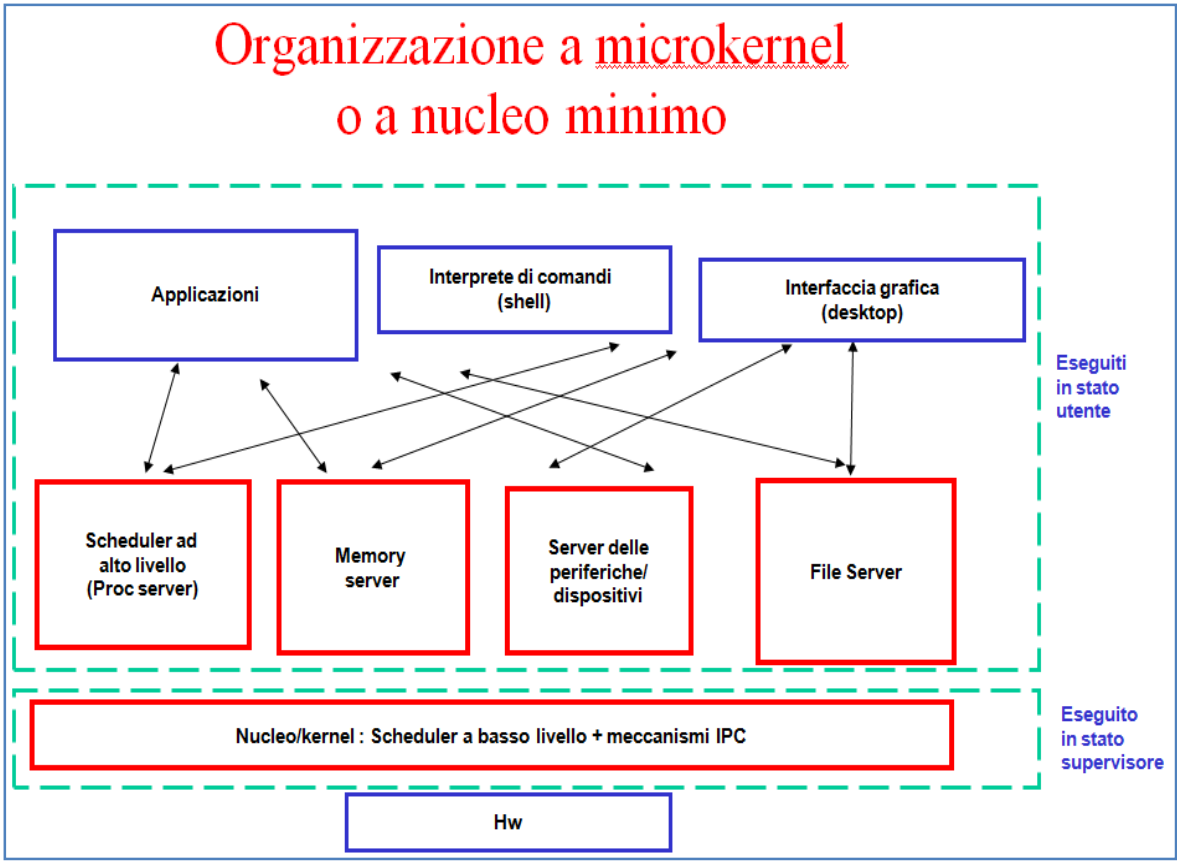

Organizzazione a microkernel
o a nucleo minimo
Applicazioni
Interprete di comandi (shell)
Interfaccia grafica (desktop)
Eseguiti in stato utente
Scheduler ad alto livello (Proc server)
Memory server
Server delle periferiche/ dispositivi
File Server
Nucleo/kernel : Scheduler a basso livello + meccanismi IPC
Eseguito in stato supervisore
Hw

Fig.3

Gli strati del Sistema Operativo

Presentiamo in sintesi le principali funzioni svolte dal sistema operativo. La Fig.1 descrive l'organizzazione a strati del sistema operativo. Andando dal basso verso l'alto, esso comprende cinque strati.

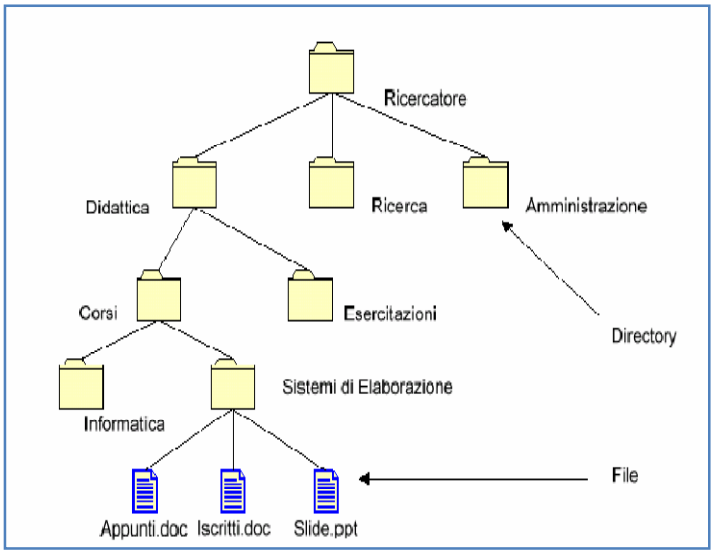
- **Il gestore dei processi** (è la parte più importante del kernel) è responsabile dell'***esecuzione dei programmi*** da parte dell'unità di elaborazione (**CPU**).
  In caso siano presenti molti utenti, ciascuno interessato all'esecuzione di un particolare programma, il nucleo deve garantire l'esecuzione quasi contemporanea di molti processi; perciò deve decidere a quale di essi assegnare l'accesso e l'uso dell'unità di elaborazione. Inoltre, il gestore dei processi è responsabile di reagire agli eventi esterni all'unità centrale (in pratica, segnali provenienti dalle periferiche), che indicano come il calcolatore comunica con il mondo esterno. Questo strato offre agli strati superiori una macchina virtuale in cui ciascun programma opera come se avesse a disposizione un'unità di elaborazione dedicata. Anche quando un sistema ha più di un processore, questi vengono normalmente condivisi fra molti più utenti, per cui tutte le CPU devono essere gestite da questa componente del sistema operativo.

- **Il gestore della memoria** ha la funzione di **allocare la memoria centrale** (**RAM**) e partizionarla tra i vari programmi che la richiedono. Infatti, in un sistema multiutente è opportuno che molti programmi siano contemporaneamente presenti in memoria, in modo che possa aver luogo la loro pseudo-simultanea esecuzione. Questo strato offre agli strati superiori una macchina virtuale in cui ciascun programma opera come se avesse disponibile una memoria dedicata.

- **Gestore delle periferiche (driver)** è responsabile delle **operazioni di ingresso/uscita** che coinvolgono le periferiche; i driver sono programmi che eseguono operazioni di ingresso/uscita per uno specifico componente. Questo strato offre all'utente una visione astratta in cui le caratteristiche hardware delle periferiche vengono mascherate; l'utente ha a disposizione un insieme di procedure standard di alto livello, che leggono dati in ingresso (es. tastiera, mouse, hard disk, …) e scrivono dati in uscita (monitor, stampante, hard disk, ...). L'utente, anche in questo caso, ha l'impressione che la periferica sia dedicata.

  I SO comprendono i driver per la gestione delle periferiche più comuni. Ogni aggiunta o modifica alla configurazione standard comporta l'installazione di software addizionali (driver aggiuntivi).
  I SO più recenti sono dotati di funzioni Plug&Play che permettono di aggiungere (plug) nuove periferiche al sistema che possono essere utilizzate (play) senza l'intervento dell'utente per la selezione e l'installazione del driver.

- **File System (gestore delle informazioni)** è responsabile della gestione dei file in memoria di massa; esso struttura i dati in **file**, li organizza in **directory** (o **cartelle**), ossia contenitori di file, e fornisce all'utente un insieme di funzioni di alto livello per operare su di essi, mascherando le operazioni che vengono realmente effettuate per allocare la memoria di massa e per accedervi in lettura o scrittura. La struttura delle cartelle e sottocartelle prevede una struttura ad albero. Tramite il file System, ciascun utente può organizzarsi una zona della memoria di massa e garantire che i suoi file siano adeguatamente protetti da accessi esterni; il file system consente anche che alcuni file vengano condivisi fra più utenti.



  Il file rappresenta l'unità minima di archiviazione ed è caratterizzato da: un **nome**, un **contenuto** (informazioni memorizzate) e degli **attributi** (dimensione, data di creazione, diritti di accesso).

- **L'interprete dei comandi** consente all'utente di attivare i programmi (per esempio, mandare in esecuzione un elaboratore di testi); per eseguire un programma, l'interprete dei comandi svolge in modo invisibile all'utente un insieme di operazioni, tra cui:
  - accedere al programma, tipicamente residente in memoria di massa, tramite il file system;
  - allocare memoria centrale e caricarvi il programma e i relativi dati, tramite il gestore della memoria;
  - creare e attivare il processo o i processi corrispondenti al programma da eseguire, tramite il gestore dei processi.

  Si noti che l'interprete comandi sfrutta l'organizzazione a strati del sistema operativo, cioè può richiedere l'esecuzione di tutte le funzioni di più basso livello rispetto a esso. In alcuni casi, l'interprete comandi non è incluso nel sistema operativo, bensì nel livello più basso dell'applicazione, detto gestore delle applicazioni.

Il SO cerca di fare delle scelte convenienti per gli utenti; quando essi siedono a uno dei tanti terminali e richiedono servizi al sistema; per esempio, in un sistema di gestione dei conti correnti in una banca, l'obiettivo consiste spesso nel ridurre i tempi di attesa al terminale. Il sistema operativo cerca di ottimizzare le prestazioni del sistema informatico, determinando le politiche migliori di gestione delle risorse sotto il suo controllo. Un utente risente della presenza degli altri utenti in misura crescente con il **carico complessivo** del sistema informatico, cioè con il crescere del numero complessivo di richieste di elaborazione fatte da tutti gli utenti.