

Indice

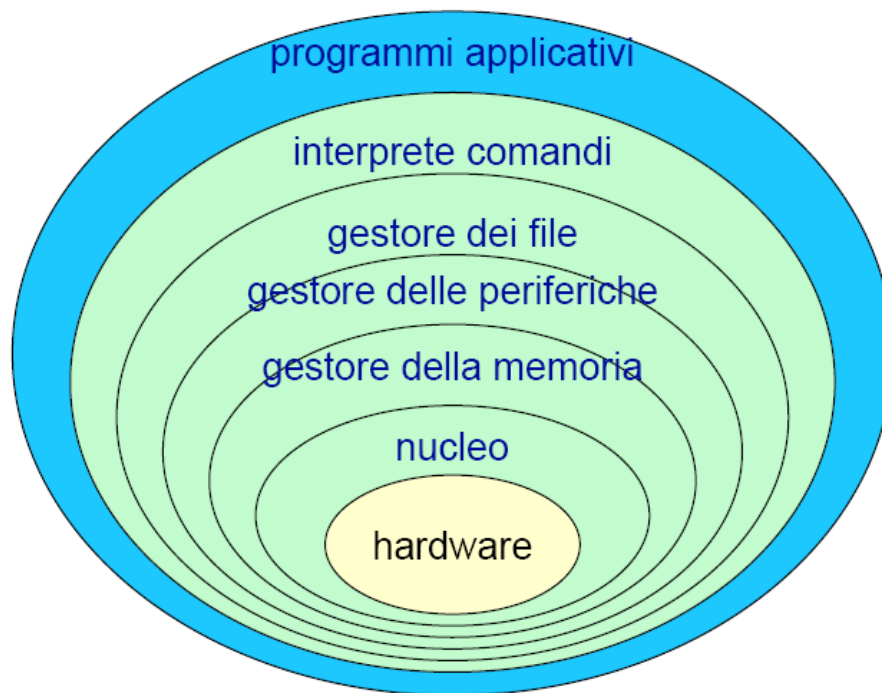
Compiti del sistema operativo.....	1
Architettura a livelli di un sistema operativo.....	2
Nucleo.....	4
Gestore della memoria.....	6
Gestore delle periferiche.....	9
Gestore dei file.....	10
Interprete comandi.....	15
Programmi di utilità.....	18
Categorie di Sistemi Operativi.....	19
Architetture Client Server.....	20

Compiti del sistema operativo

Il **sistema operativo** è il SW di base responsabile della gestione delle risorse del calcolatore, e in particolare di come le risorse vengono allocate alle applicazioni, ad esempio, l'utente può richiedere al sistema operativo di eseguire un programma

- facendo un doppio click sull'icona di un programma sul desktop del calcolatore
- per avviare l'esecuzione del programma, il sistema operativo svolge le seguenti attività:
 - individua il codice eseguibile del programma (memorizzato sul disco)
 - alloca al programma le risorse necessarie per la sua esecuzione (ad esempio, una certa quantità di memoria centrale)
 - carica il codice eseguibile del programma in memoria centrale
 - avvia il programma

Architettura a livelli di un sistema operativo



I sistemi operativi hanno una struttura complessa, che può essere descritta come una gerarchia di macchine virtuali

Ciascun livello della gerarchia è relativo a una macchina virtuale del sistema operativo che gestisce una diversa tipologia di risorse le risorse gestite sono i processori, la memoria, le periferiche, le memorie secondarie, l'interfaccia utente

N.B.:

l'utente del calcolatore interagisce solo con l'interprete comandi del sistema operativo

Nucleo

Il **nucleo** del sistema operativo (o **gestore dei processi**) è responsabile della gestione della risorsa unità di elaborazione (processore) sotto il nucleo esiste fisicamente una sola unità di elaborazione quindi il calcolatore potrebbe svolgere un solo programma alla volta

il nucleo ha (tra l'altro) lo scopo di definire tante unità di elaborazione virtuali, una per ciascun programma che si vuole eseguire

ciascun **processo** ha l'impressione di essere l'unico programma eseguito dal calcolatore

il nucleo ripartisce la capacità di elaborazione dell'unica unità di elaborazione reale tra le tante unità di elaborazione virtuali

il nucleo gestisce anche l'eventuale presenza di più unità di elaborazione

I processi contemporanei vengono eseguiti sulla CPU per un **quanto di tempo** e poi la cedono ad un altro processo per dare all'utente l'impressione che siano tutti attivi.

Quando non sono eseguiti vengono conservati in memoria attraverso il **PCB** Process Control Block (che ne conserva lo stato e le informazioni di esecuzione).

Per fare questo si gestisce una coda dei processi che devono essere eseguiti.

Un processo può essere in uno dei seguenti 3 stati:

- **Pronto (READY):** possiede tutte le risorse ed aspetta solo che venga il suo turno di uso della CPU (dove viene gestita la coda)
- **Esecuzione (RUNNING):** è in esecuzione sulla CPU
- **Attesa (WAIT):** sta aspettando che si liberi una risorsa (accesso al disco, stampante, monitor, etc...)

Gestore della memoria

Il **gestore della memoria** consente l'allocazione dinamica della memoria centrale ai programmi in esecuzione

a ciascun programma viene allocata una area di memoria virtuale sufficiente per la sua esecuzione

Il gestore della memoria gestisce la corrispondenza tra le memorie virtuali e l'unica memoria reale

La dimensione della memoria virtuale può essere maggiore di quella reale

I dati possono essere parcheggiati temporaneamente nella memoria secondaria

Memoria Virtuale

I dati stanno in parte sul Disco ed in parte sulla RAM, soprattutto per contenere programmi o files di dati più grandi della dimensione della RAM, nei PC.

Parte del programma e dei dati strettamente necessari in RAM, il resto del programma e dei dati su area HD detta "*area di swap*"

Si ha quindi un'espansione virtuale della dimensione della Memoria non più legata alla dimensione della RAM fisica

Catena di programmazione

Un programma eseguibile è un insieme di istruzioni binarie (programma oggetto) che fanno riferimento ad indirizzi di memoria.
ad esempio somma il contenuto della cella di indirizzo 2004 con quello della cella di indirizzo 2006 e mettilo nella cella con indirizzo 2008

Visto che i programmi vengono caricati dal gestore della memoria in maniera dinamica, nel senso che la zona di memoria utilizzata è la prima libera, quindi non si conosce l'indirizzo della prima cella.

Allora il programma oggetto è generato in maniera che gli indirizzi utilizzati sono basati sul principio che il programma inizi all'indirizzo 0, questo programma si dice **rilocabile**

Il caricamento in memoria del programma quando deve essere eseguito viene gestito dal **loader** che opera questo calcolo degli indirizzi, detto **rilocalizzazione**, in 2 modalità: **Statica** (modifica indirizzi al caricamento del programma) e **Dinamica** (ricalcolo RunTime)

Gestore delle periferiche

Il **gestore delle periferiche** adatta la modalità d'uso delle singole periferiche (che possono essere estremamente diverse) a quello di poche tipologie di periferiche virtuali

ad esempio, un programma può usare una stampante senza conoscere i dettagli precisi di funzionamento della stampante fisicamente disponibile

Un **driver** è un programma che adatta i dati dal formato virtuale (generico) usato dal gestore delle periferiche a quello reale della periferica effettivamente in uso

Il Sistema Operativo conosce le periferiche generiche (stampanti, monitor, schede video, etc...) ed è in grado di gestire i driver specifici dell'HW forniti dal produttore di quell'HW.

*N.B.: i moderni sistemi operativi implementano il meccanismo del **Plug and Play** che gli consente di interrogare direttamente l'HW collegato riconoscerlo ed installarne il driver specifico*

Gestore dei file

Il **gestore dei file** (o **file system**) è dedicato alla gestione della memoria secondaria (dischi)

i dischi sono fisicamente organizzati in superfici, cilindri e settori, di dimensioni prefissate

il gestore dei file consente l'uso della memoria secondaria mediante una organizzazione logica in volumi, cartelle e file

un file è una sequenza di dati di dimensioni variabili, a cui il sistema operativo fa corrispondere una serie di aree di disco organizzate come sopra

una cartella (o direttorio) è il contenitore di un insieme di file e cartelle

un volume è un contenitore di file e cartelle, solitamente corrispondente ad una unità a disco (es. C)

inoltre, il gestore dei file mostra ai livelli superiori anche le periferiche come dei file, da cui è possibile leggere e/o su cui è possibile scrivere sequenze di dati

Gestione dei file

Il file system è il componente del sistema operativo responsabile della gestione dei dati nelle memorie secondarie

il sistema operativo gestisce la corrispondenza tra l'organizzazione logica e l'organizzazione fisica dei dati

i dati nelle memorie secondarie sono organizzati logicamente in volumi, cartelle e file

i dati nelle memorie secondarie sono organizzati fisicamente in dischi, superfici, cilindri e settori

Ad esempio in un sistema operativo tipo Windows

*Un **file** è un contenitore logico di informazioni, caratterizzato da*

- *un contenuto*
- *una sequenza di byte, da interpretare opportunamente*
- *un nome che ne consente l'identificazione*

il nome è composto dal nome del file e da un percorso

*ad esempio **C:\corso\esercizi\prova.doc***

***C:** il volume*

***C:\corso\esercizi** è il percorso*

***prova.doc:** il nome del file composto da 2 parti:*

- *prova: nome scelto dall'utente*
- *.doc: l'estensione che i Sistemi Operativi come Windows utilizzano per sapere quale programma usare per aprirlo*

Volumi e cartelle

L'insieme dei file di un calcolatore non viene gestito in un unico contenitore piatto, ma piuttosto i file vengono organizzati in volumi e cartelle un **volume** è un contenitore logico di file, corrispondente solitamente a una unità a disco

*ad esempio, il volume **a:** indica il floppy disk e **c:** il disco fisso primario*

una **cartella** (o **directory**) è un contenitore logico di file e cartelle le cartelle sono organizzate gerarchicamente ad albero

- ogni cartella può contenere dei file
- ogni cartella può contenere altre cartelle (chiamate sottodirettori)
- la radice (root) è la cartella di livello gerarchico più elevato, corrispondente a un intero volume

L'applicazione **Explorer** (Gestione risorse) di Windows consente la gestione del file system mediante una interfaccia grafica consente la visualizzazione del contenuto di volumi e cartelle, nonché della loro organizzazione gerarchica

Interprete comandi

L'**interprete comandi** è l'interfaccia utente del sistema operativo definisce le operazioni che possono essere utilizzate direttamente dall'utente finale, chiamate **comandi**

Esempi di comandi sono

- esecuzione di una applicazione
- apertura di un documento (eseguendo l'applicazione responsabile della manipolazione di quel tipo di documento)

L'interprete comandi può essere:

- basato su una interfaccia a caratteri (come nei sistemi operativi Unix, ma anche nel "prompt dei comandi" di Windows)
- basato su una interfaccia grafica (come nei sistemi operativi Windows e MacOS e nell'ambiente X Window di Unix)

Evoluzione delle Interfacce dei S.O.

L'evoluzione del S.O. è andata molto, negli ultimi anni, nella direzione di cercare di dare all'utente un'interfaccia sempre più semplificata e sofisticata per utilizzare il Computer.

Gli stadi principali di questa evoluzione dell'interfaccia sono stati:

Batch: si predisponavano le schede con programmi e dati e si davano in pasto al calcolatore (anni 60-70).

A comandi: il S.O. aveva un set di comandi che l'utente gli poteva dare e che interpretava (MS-DOS, UNIX, MVS-IBM, etc...) (anni 70-90, ma ancora presente sotto molti S.O.)

A menù: il S.O. metteva a disposizione dell'utente un insieme di menù attraverso il quale questi poteva navigare per svolgere le sue attività (DOS-Shell, etc...) (anni 80-90, ma ancora presente).

GUI: il S.O. offre una serie di icone a cui sono associate le azioni e che l'utente deve solo cliccare (Windows, MacOS, X-Windows, etc...) (anni 80 fino ad oggi).

GUI (Graphical User Interface)

Le sue funzionalità principali sono:

- Visualizza sullo schermo le finestre corrispondenti a programmi attivi
- WYSIWYG (What You See Is What You Get)
- Stessi comandi per tutti i programmi (cancella, copia, incolla, salva, etc...)
- Presenta i comandi come Icone

Programmi di utilità

Il sistema operativo è corredato da un insieme di programmi di utilità, per la configurazione, l'ottimizzazione e la gestione ordinaria del sistema:

- gestione degli utenti
 - per definire gli utenti, i gruppi di utenti e i diritti di accesso alle varie risorse
- gestione delle periferiche
 - ad esempio, l'utilità di formattazione dei dischi e l'utilità di deframmentazione
- gestione delle applicazioni
 - per l'installazione e la disinstallazione di programmi
- altri programmi accessori
 - dall'editor di testi alla gestione dei servizi Internet

Categorie di Sistemi Operativi

- **Sistemi dedicati** specializzati in una sola attività come console di video giochi, decoder, etc..., hanno tutte le funzionalità specializzate
- **Sistemi Batch** specializzati per attività che richiedono soprattutto tempi lunghi e poche interazioni con l'utente
- **Sistemi Time Sharing** prevedono più processi contemporanei dando ad ogni processo e/o utente l'impressione di avere una macchina intera a disposizione.
- **Sistemi Real Time** specializzati per attività che richiedono la risposta immediata da parte del calcolatore, ad esempio quelli per controllo processi e/o impianti
- **Sistemi Transazionali** orientati a gestire proprio transazioni, cioè operazioni articolate su più passi, come ad esempio un'operazione bancaria

Architetture Client Server

Più che una tipologia di sistema operativo si parla di un'architettura applicativa, dove il carico elaborativo è suddiviso tra 2 sistemi. Infatti il Server offre una serie di servizi e si occupa di una parte di elaborazione mentre il Client si occupa dell'altra parte.

Un esempio interessante e caratteristico possono essere la **posta elettronica**, dove

- **Server:** gestisce la ricezione, la spedizione dei messaggi e la conservazione degli stessi
- **Client:** gestisce la parte di interazione con l'utente, la lettura del messaggio, la composizione, l'invio e la ricezione da e verso il Server

Ed anche il Web, dove il Server conserva le pagine HTML (e le immagini, etc...) ed il Client li visualizza.