Interfaccia

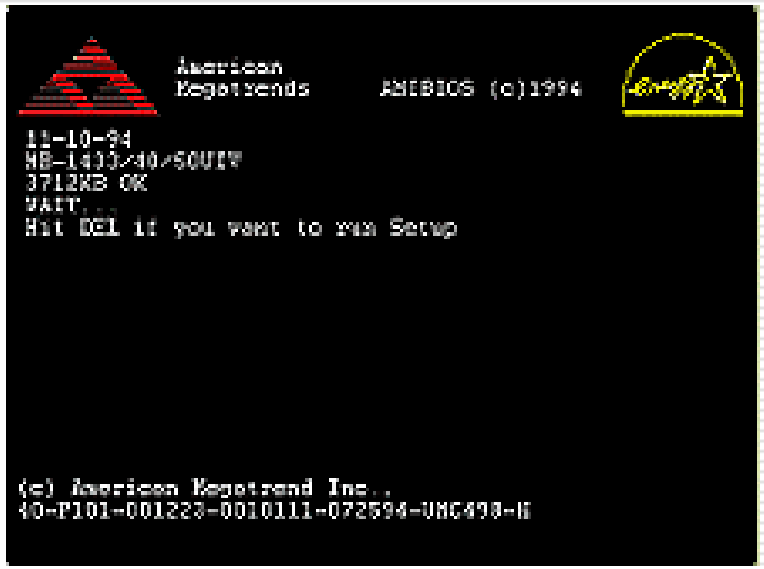
Due sono i tipi di interfaccia più diffusi per interagire con un computer:

- Interfacce **a caratteri**, anche dette a **riga di comando** (*CLI, command line interfaces*)

- Interfacce **grafiche** (GUI, graphical user interfaces), in particolare di tipo WIMP (Windows, Icons, Menus, Pointing device)

Sistemi Operativi: avvio

All'avvio del computer, terminate le verifiche del BIOS, il controllo passa al **sistema operativo**.



Il Sistema Operativo opera come **intermediario** tra l'hardware del sistema e uno o più utenti.

Sistema Operativo: funzioni

Due sono le funzioni principali di un sistema operativo:

- gestione delle risorse hardware
- interfaccia verso l'utente

Come tali funzioni vengano svolte, e se sia **interamente** il sistema operativo ad occuparsene, dipende dal dispositivo in questione.

Sistemi operativi: storia (UNIX)

Nella seconda metà degli anni ’60, la AT&T abbandona il progetto MULTICS – un sistema a time-sharing – rivelatosi deludente.

Ken Thompson, allora ai Bell Labs della AT&T, scrive un sistema operativo in assembler e lo battezza, scherzosamente, UNICS (Uniplexed Information and Computing Services).

Nasceva così **UNIX**, capostipite di una numerosa e varia famiglia di sistemi operativi. Era il 1969.

Sistemi operativi: storia (UNIX)

Nel frattempo **Dennis Ritchie**, che collaborava con Thompson, sviluppa il linguaggio di programmazione “C”.

Nel 1972 UNIX viene riscritto in C, aprendo la strada all’utilizzo su *altre piattaforme hardware*.

Dal 1975 in poi, UNIX inizia a diffondersi nelle **università** – negli Stati Uniti, in Giappone, Australia, Europa…

Sistemi operativi: storia (DOS)

Negli anni ’70, iniziano a diffondersi i primi personal computer. **Gary Kildall** sviluppa, nel 1972, il **CP/M** (Control Program for Microprocessors), un sistema operativo funzionante su uno dei primi processori Intel (il 4004).

Il **primo BIOS** fu scritto da Kildall come modulo per il proprio CP/M, in modo da rendere il sistema meno dipendente dall’hardware.

Sistemi operativi: storia (DOS)

Nel 1980 **Tim Patterson**, della Seattle Computer Products, sviluppò un proprio sistema operativo basato sul CP/M. Impiegò due mesi, e lo battezzò **QDOS** (Quick and Dirty Operating System).

Dopo quattro mesi, un'altra ditta di Seattle, la **Microsoft** di *Bill Gates*, acquista dalla SCP i diritti per rivendere il DOS ad un cliente non menzionato.

Il cliente è **l'IBM**, che nel 1981 lancerà il primo PC, dando il via alla rivoluzione dei personal computer.

Sistemi operativi: storia (Win & Mac)

La *Apple Computer*, negli stessi anni, sta portando avanti diversi progetti, e sperimenta diversi sistemi operativi. Nel 1984, con il lancio (e il successo) dell'**Apple Macintosh**, si focalizzerà sul suo sistema operativo, il **System** – allora alla versione 1.0.

Al contrario del DOS, nelle sue varie incarnazioni, il System ha un'interfaccia grafica.

L'anno successivo, il 1985, la Microsoft lancia la prima versione di **Windows**.

Sistemi operativi: storia (Linux)

Nel 1991 uno studente finlandese, **Linus Torvalds**, sviluppò il kernel per un sistema operativo basato su una variante di UNIX.

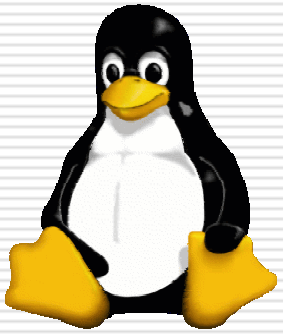
Lo distribuì in rete secondo la GNU *General Public License*, una licenza che ne consentiva l'uso, la redistribuzione e la modifica (a certe condizioni).

Iniziò a ricevere da subito contributi da altri sviluppatori.

Sistemi operativi: storia (Linux)

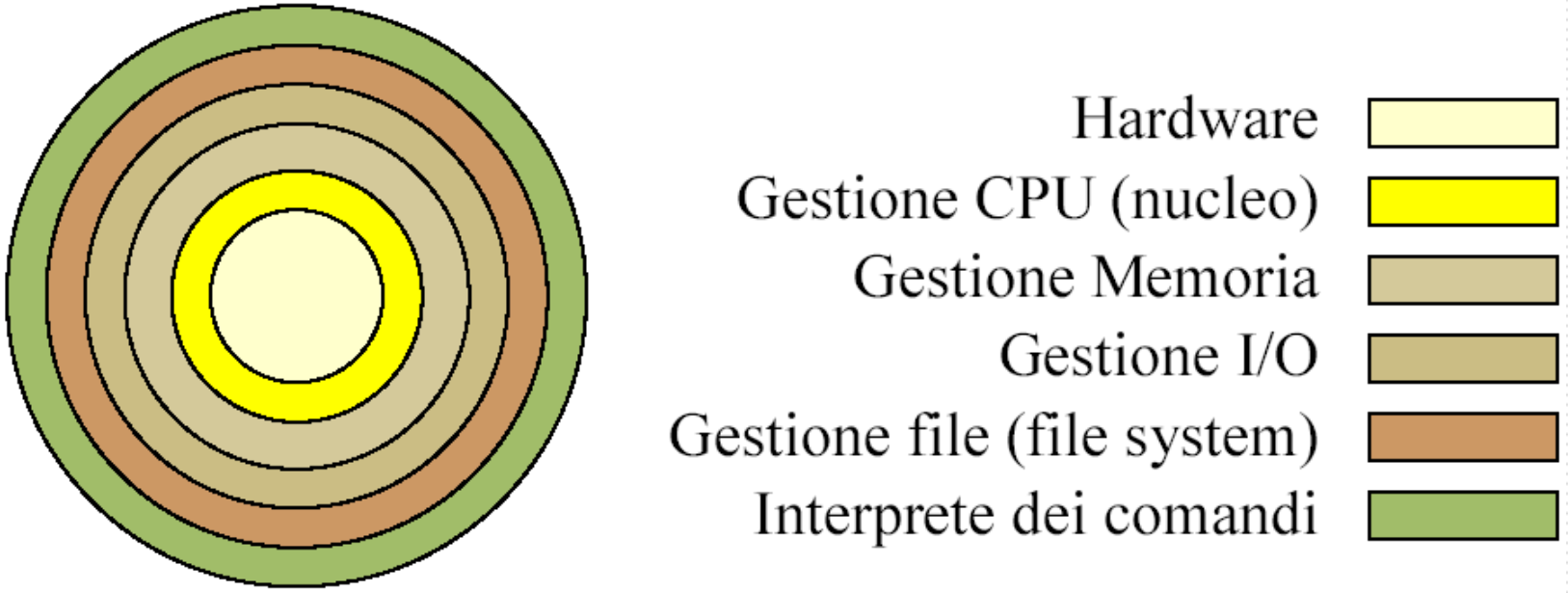
Nasceva così **Linux**, una delle varianti UNIX oggi più diffuse. Il kernel di Linux è continuamente aggiornato, e disponibile anche gratuitamente.

Uno dei punti di forza di questo sistema è la **comunità** che lo supporta, e la filosofia su cui si basa, quella del **software libero**.



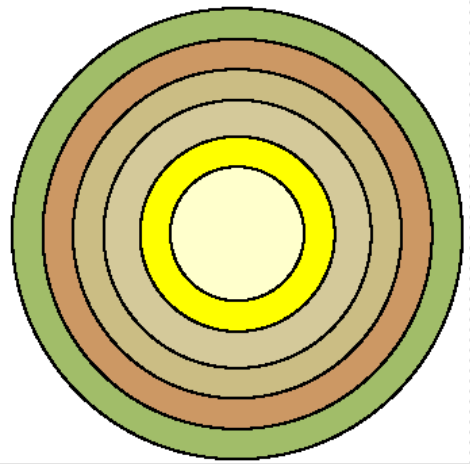
livelli

La **struttura** di un sistema operativo, tipicamente, è quella *a cipolla*:



livelli

L'hardware è dunque "ricoperto" da una serie di **strati** di software.



Ciascun livello:

– usa le **funzionalità** di quello sottostante

– fornisce **servizi** al livello che segue nella gerarchia

– gestisce delle **risorse** mediante politiche invisibili ai livelli superiori

COMPITI GENERALI

Gestione CPU

Gestione memoria

Gestione I/O

Gestione files (file system)

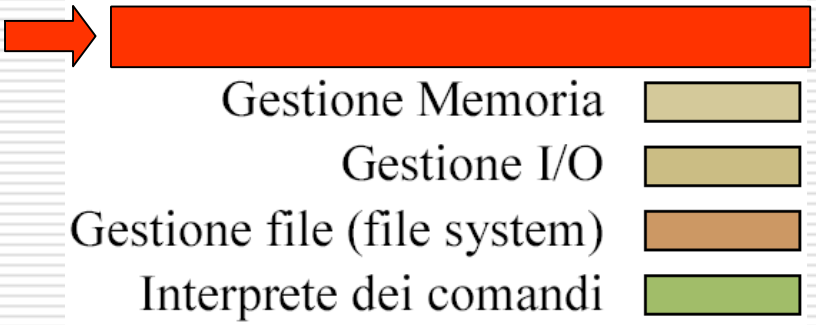
Interprete dei comandi

gestione CPU

Il livello più basso è quello del **kernel** (nucleo). Questa parte del sistema operativo si occupa di gestire l'esecuzione dei programmi.

Un programma in esecuzione è detto **processo**.

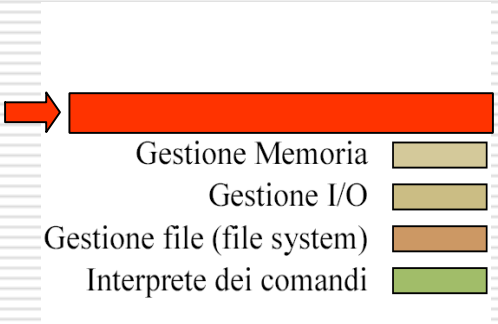
Il kernel **distribuisce** le risorse di calcolo tra i vari processi attivi.



gestione CPU

Una prima distinzione è dunque tra quei sistemi che eseguono un processo per volta e quelli in grado di gestirne diversi “contemporaneamente”.

Questi ultimi sono detti **multitasking**.

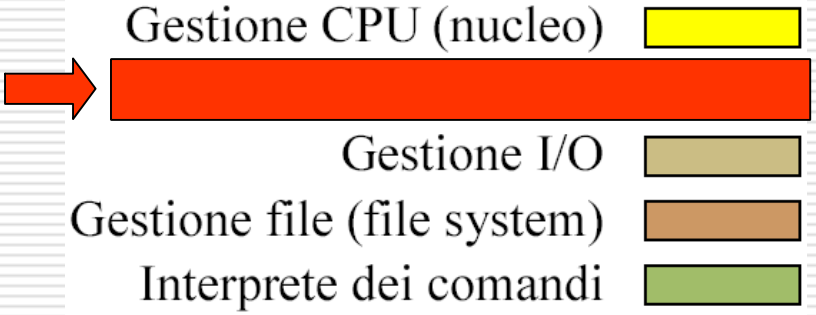


gestione memoria

La memoria è una risorsa **essenziale** e **limitata**.

Essenziale, perché ogni programma in esecuzione (processo) deve essere “caricato” in memoria, e così i dati su cui opera.

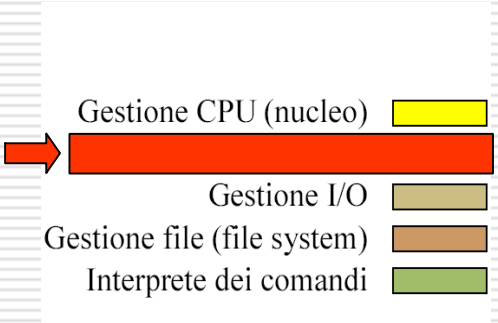
Limitata, perché nei sistemi moderni possono essere attivi più processi nello stesso tempo.



gestione memoria

Dal momento che la memoria di sistema (RAM) è una risorsa finita, nell'allocarla ai vari processi il sistema operativo deve risolvere vari problemi:

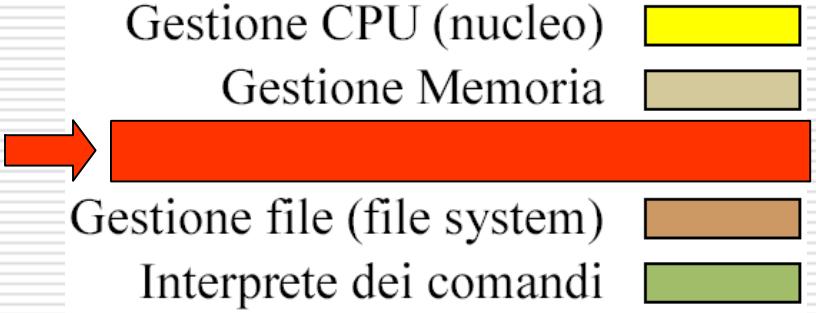
- trovare **spazio** per i vari processi;
- "**rilocare**" il codice caricato in memoria;
- ridurre la **frammentazione.**



Gestione input/output (I/O)

L'**accesso** alle periferiche di I/O viene gestito dal sistema operativo insieme ai *driver di periferica*.

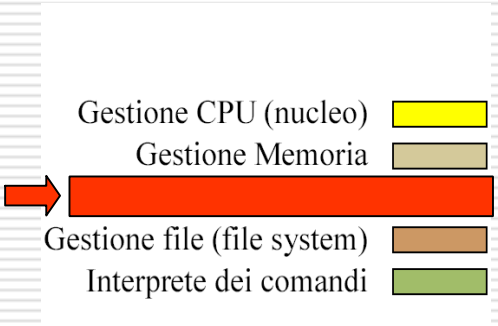
Questi sono programmi specifici per ciascun dispositivo che si colleghi all'elaboratore (stampanti, scanner dischi…).



Gestione input/output (I/O)

L'interazione tra un programma e una periferica è standardizzata. Un programma di elaborazione testi, ad esempio, può inviare un comando di stampa senza curarsi del tipo di stampante collegata al computer.

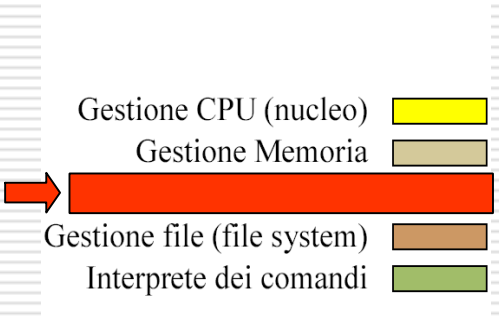
Spetta al sistema operativo smistare la richiesta al driver della stampante.



Gestione input/output (I/O)

A questo livello è implementato anche un sistema di gestione degli **errori di I/O** (ad es. dischetto mancante o danneggiato, carta esaurita, ecc.).

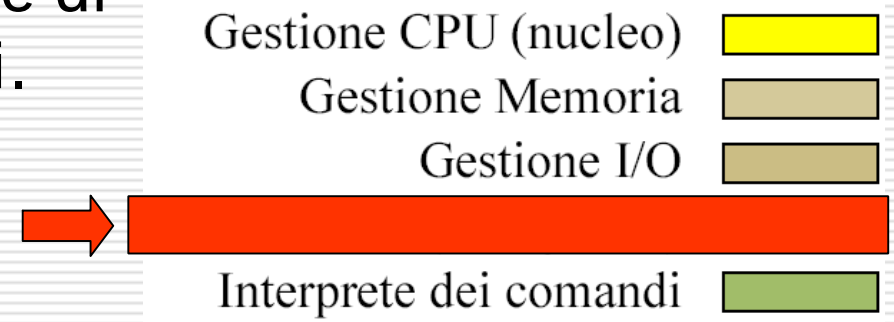
Anche il controllo dell'ordine di accesso ai dispositivi è cruciale. Il sistema operativo deve prevenire, o risolvere, eventuali conflitti.



Gestione files

Il **file system** è il modo in cui il sistema operativo organizza i file (documenti) sulle unità di memorizzazione.

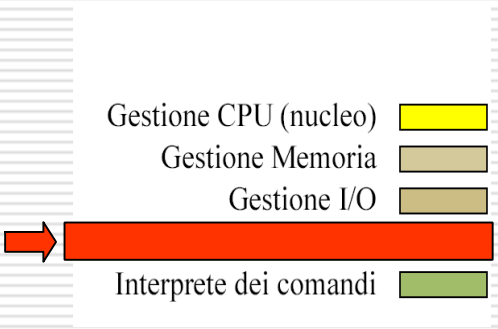
Un **file** è un'astrazione che rappresenta un insieme di byte logicamente collegati.



Gestione files: funzioni

Il **file system** deve mettere a disposizione diverse funzioni per la manipolazione dei file:

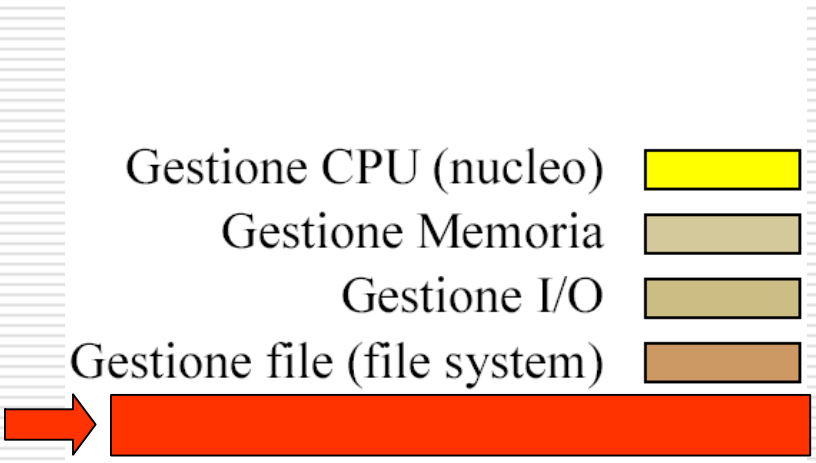
- **creazione/eliminazione**
- **lettura/scrittura/esecuzione**
- **coordinamento** accessi contemporanei
- **controllo** degli accessi (nei sistemi multiutente)



Interprete dei comandi (shell)

L'**interprete dei comandi** è quella parte del sistema operativo che riceve ed elabora le istruzioni impartite da un utente.

E' possibile utilizzare lo stesso sistema operativo con **shell** differenti. Questo può rendere molto diverso il modo di impartire comandi.

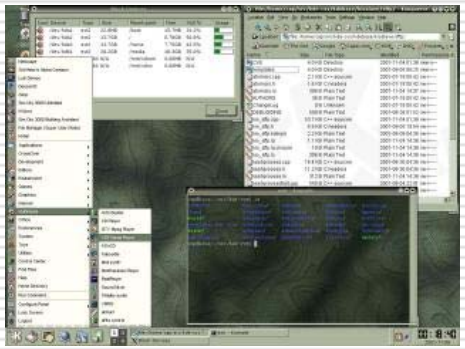


Interprete dei comandi (shell)

La shell è dunque lo strato più esterno di un sistema operativo. Di fatto, rappresenta l'**interfaccia** tra utente e sistema.

MacOS X

Windows XP



Linux - KDE 3.0