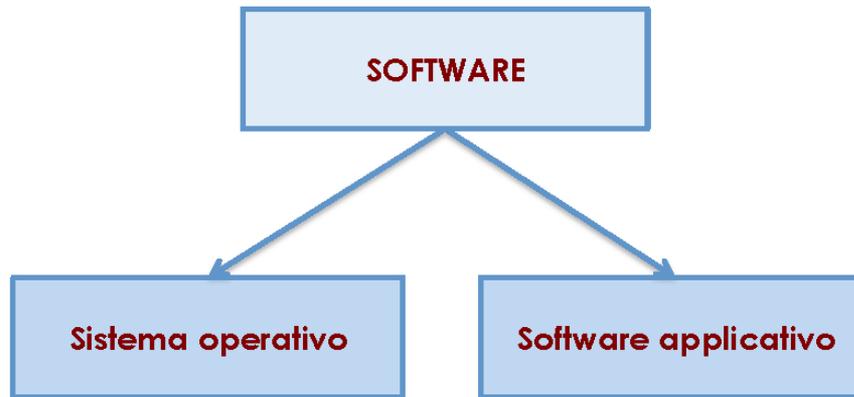


Classificazione del **software**

Sulla base del loro utilizzo, i programmi si distinguono in:



Un **sistema operativo** è un complesso insieme di componenti software che ha il compito di gestire l'esecuzione dei programmi applicativi, garantendo un utilizzo, **facile, efficiente** e **sicuro** delle risorse fisiche che la macchina mette a disposizione. E' un **intermediario** fra i dispositivi che costituiscono un sistema di elaborazione e i programmi degli utenti: nasconde i molteplici e complessi dettagli operativi interni e realizza al di sopra della “**macchina reale**” una “**macchina virtuale**” utilizzabile con relativa facilità mediante un insieme organico di comandi.



Si è visto che a livello fisico un sistema di elaborazione è costituito da un insieme di dispositivi elettronici, elettro-meccanici, magnetici e ottici destinati a realizzare delle funzioni fondamentali.

Per mezzo di questi dispositivi, denominati **hardware**, è possibile eseguire programmi dedicati al soddisfacimento delle esigenze degli utenti, indicati con il termine **software**

Sebbene hardware e software operino a livelli distinti nella struttura di elaborazione (HW a livelli di tensioni, correnti, stati magnetizzati, il SW a livello di istruzioni) un programmatore di applicativi non "dovrebbe" prescindere dalla conoscenza specifica del modo di operare dei dispositivi fisici

Esempio:

Un programmatore di applicativi sta progettando un software di videoscrittura, dove all'interno del quale è previsto la lettura dei dati attraverso un lettore DVD

Il programmatore dovrà prevedere delle specifiche apposite che operino a livello fisico della macchina (interagire direttamente con il lettore DVD, conoscere gli indirizzi di memoria della memoria di massa, etc...)

Questo aspetto è impraticabile almeno per due motivi fondamentali:

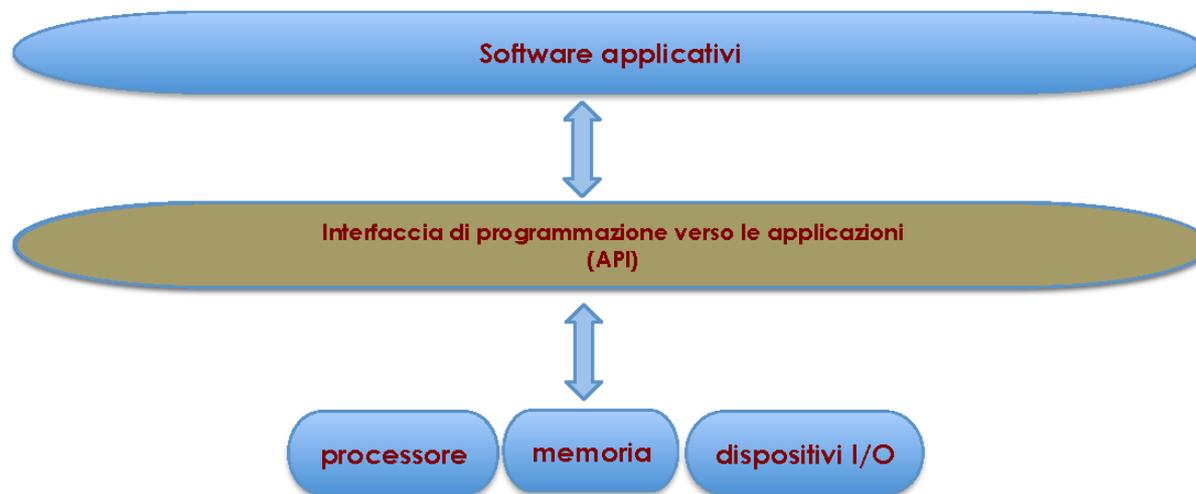
- ❑ renderebbe la scrittura di programmi particolarmente complessa e pesante
- ❑ legherebbe il programma scritto, allo specifico sistema hardware per il quale è stato scritto.

Per far fronte a questi problemi risulta opportuno suddividere il software di un sistema di elaborazione in due livelli distinti

- **Sistema operativo** (*operating system*) che comprende i programmi dedicati alla gestione del livello fisico delle macchine;
- **Applicazioni** (*application software*) che comprendono i programmi destinati al soddisfacimento delle esigenze specifiche degli utenti.

Una suddivisione di questo tipo fa sì che le **applicazioni** interagiscono con l'hardware della macchina solo indirettamente, tramite **l'intermediazione** del **sistema operativo**

In questo modo:



L'interfaccia che il S.O. mette a disposizione delle applicazioni è detta **interfacce di programmazione verso le applicazioni** o **API** (*Application Programming Interface*) sono un insieme di comandi o primitive, dette **chiamate di sistema** (system call), che posso essere utilizzate dal programmatore delle applicazioni come se fossero normali istruzioni in linguaggio macchina

Interfaccia di programmazione verso le applicazioni
(API)

Questi comandi messi a disposizione fanno sì che il programmatore delle applicazioni non necessariamente conosca il linguaggio del sistema fisico, sopperendo così a quella complessità descritta precedentemente.

Inoltre definiamo **Kernel** quella parte del sistema operativo che si colloca immediatamente sopra all'hardware, con il compito di "nascondere" ai livelli sovrastanti i dettagli più complessi e fungere da gestore delle risorse

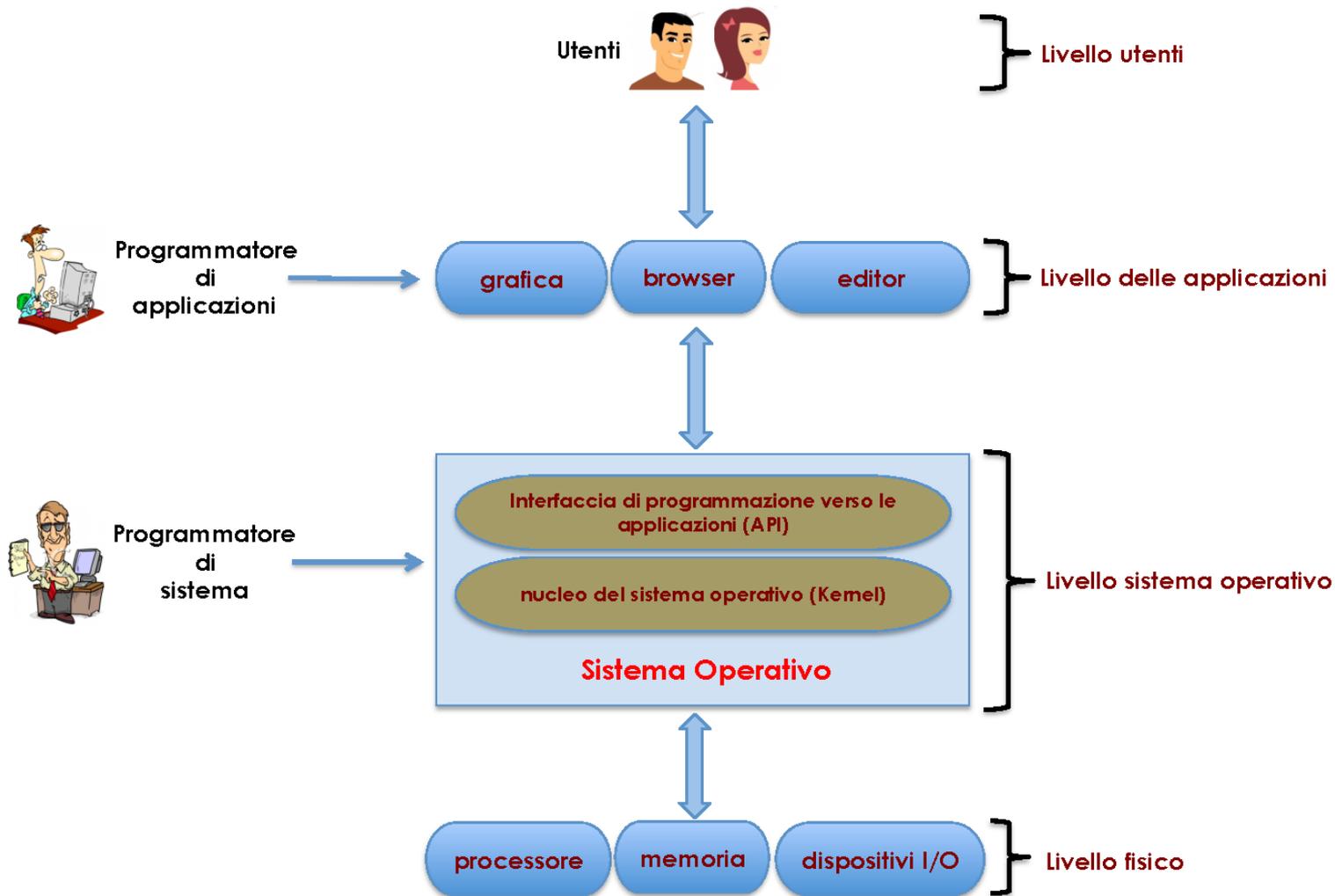


Esempio:

Un programmatore di applicativi sta progettando un software di videoscrittura, dove all'interno del quale è previsto la lettura dei dati attraverso un lettore DVD

Il programmatore non dovrà prevedere più delle specifiche apposite che operino a livello fisico della macchina, ma basti che usi i comandi messi a disposizione delle **API** per impartire i comandi apposti

Organizzazione a livelli di un sistema di elaborazione



Il **sistema operativo**, sviluppato dai **programmatori di sistema**, opera direttamente al di sopra dell'hardware del calcolatore e costituisce l'interfaccia della macchina fisica verso i **programmatori di applicazioni**



Le **applicazioni** costituiscono un intermediario fra il sistema di elaborazione e gli **utenti**. Questi non accedono direttamente né al sistema operativo né tantomeno al sistema fisico, ma solo alle specifiche interfacce che le diverse applicazioni rendono disponibili



i **programmatori di applicazioni** non necessitano di una conoscenza diretta dell'hardware, ma solo la conoscenza delle API oltre al linguaggio di programmazione.



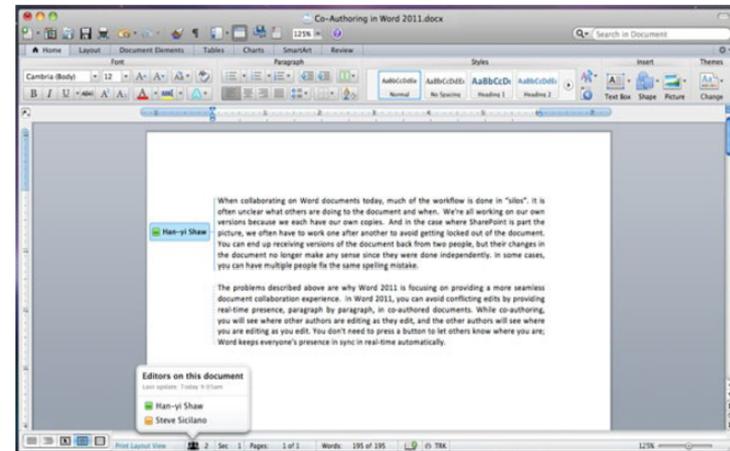
i **programmatori di sistema** necessitano una conoscenza diretta dell'hardware

Un **utente** di un sistema di videoscrittura vede il calcolatore solo attraverso l'interfaccia dell'applicazione specifica (le finestre, i menù, i comandi, le icone) e mediante la quale è in grado di scrivere testi e controllare il formato stampa senza doversi preoccupare di quanto accade al di sotto del livello al quale opera.

“**macchina**” per l'utente



Utente



In questo senso, ogni livello al di sopra della macchina fisica, definisce una nuova “**macchina**”, più **semplice**, **efficace** e **sicura** per la classe di utenti alla quale si rivolge

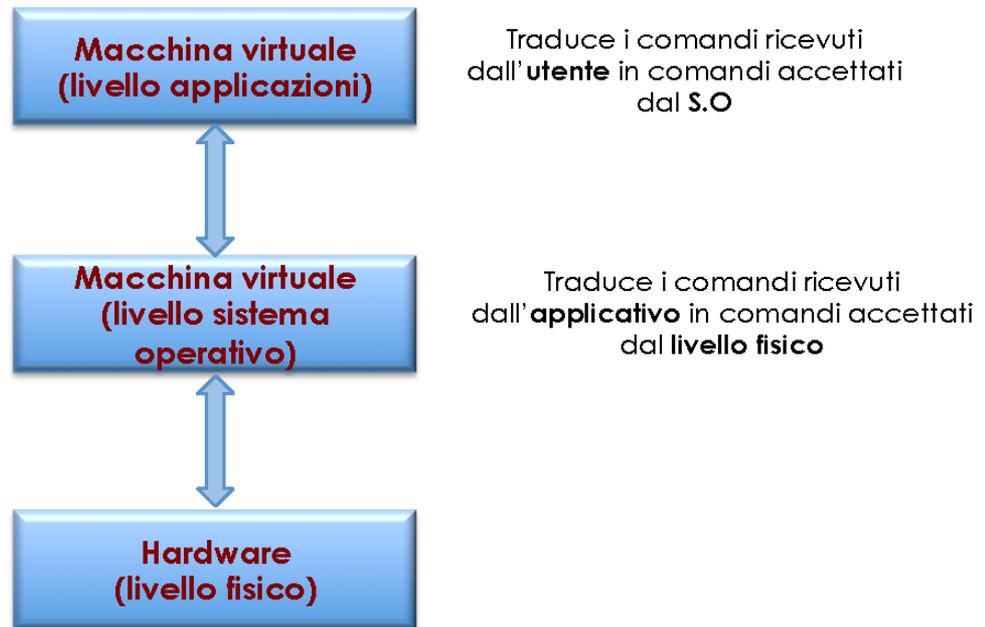
Le macchine realizzate dal sistema operativo o dalle applicazioni non sono ovviamente macchine fisiche, anche se sono del tutto reali: accettano un insieme finito di comandi e rispondono a tali comandi con azioni ben precise.



Per tali motivi, le macchine che i livelli software realizzano al di sopra della macchina fisica, prendono il nome di **macchine virtuali**. Essa è costituita da:

- ❑ **Interfaccia** attraverso la quale gli utenti della macchina virtuale possono interagire con essa, utilizzando un insieme definito di comandi.
- ❑ **corpo** interno alla macchina e non visibile dall'esterno, in grado di eseguire i comandi ricevuti utilizzando a sua volta i servizi offerti da una macchina sottostante

Ogni macchina virtuale si appoggia quindi su una macchina sottostante, in un certo senso non è altro che un “*traduttore*” che traduce i comandi ricevuti (espressi nel proprio linguaggio) in comandi accettati da una macchina di livello inferiore.

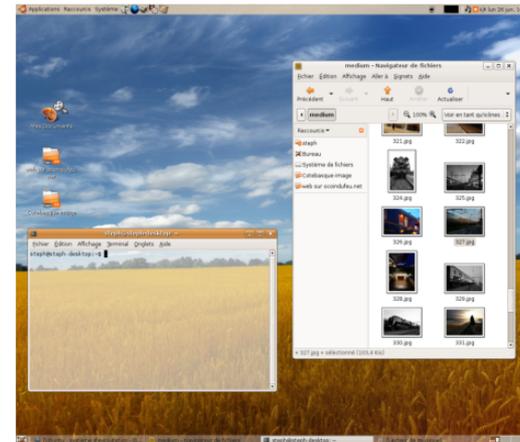
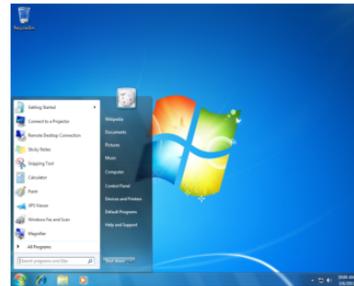


La macchina di livello più basso è la macchina fisica che, finalmente esegue direttamente i comandi ricevuti (le istruzioni macchina) per mezzo delle risorse hardware del calcolatore.

Il livello delle applicazioni comprende una grande varietà di programmi

- ❑ di videoscrittura per elaborazione testi e gestione di fogli elettronici
- ❑ di grafica, per l'elaborazione di immagini e/o disegni
- ❑ Per la gestione di grandi quantità di dati (archivi e database)

A questo livello appartiene l'interfaccia testuale e grafica del sistema operativo verso gli utenti detta **interprete dei comandi (shell)**



Osservazione

L'interprete dei comandi (shell) è confuso spesso con il sistema operativo.



I comandi, ben noti a molti utenti, che permettono di:

- ✓ organizzare i file in cartelle
- ✓ eliminare file
- ✓ spegnere il sistema
- ✓ mettere in esecuzione un programma

non sono il **sistema operativo** e nemmeno l'**interfaccia di programmazione verso le applicazioni** (API) ma semplicemente un'applicazione che permette di interfacciare il sistema operativo verso i propri utenti.

Un **sistema operativo** è un complesso insieme di componenti software che ha il compito di gestire l'esecuzione dei programmi applicativi, garantendo un utilizzo, **facile, efficiente** e **sicuro** delle risorse fisiche che la macchina mette a disposizione. E' un intermediario fra i dispositivi che costituiscono un sistema di elaborazione e i programmi degli utenti: nasconde i molteplici e complessi dettagli operativi interni e realizza al di sopra della “**macchina reale**” una “**macchina virtuale**” utilizzabile con relativa facilità mediante un insieme organico di comandi.



I nuovi sistemi operativi, presentano al loro interno un'elevata complessità strutturale. Al loro interno è possibile distinguere un certo numero di unità operative che si prendono carico di fornire uno o più servizi a programmatori e utenti

- ❑ **Gestore dei processi**
- ❑ **Gestore della memoria principale**
- ❑ **Gestore dei dispositivi Input-output (I/O)**
- ❑ **Gestore del file system**
- ❑ **Meccanismi di protezione (gestione della sicurezza)**

Ricordiamo che un **processo** rappresenta il modo in cui un programma viene eseguito nel tempo mentre il **programma** è un'entità statica data da un insieme di istruzioni in cui non specifica la distribuzione nel tempo dell'esecuzione.

Il **gestore dei processi** è l'elemento che si occupa di organizzare l'esecuzione dei programmi sul processore. Periodicamente o in seguito ad operazioni di I/O, il gestore dei processi sceglie fra i processi pronti per l'esecuzione e istruisce il processore affinché venga eseguito.

Proprio in base alla suddivisione del periodo di gestione dei processi il sistema operativo può lavorare in:

Multiprogrammazione

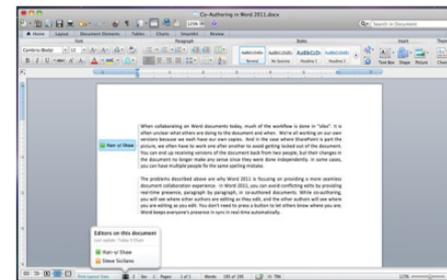
Time sharing

Si dice che un sistema operativo lavori in **multiprogrammazione** quando durante i periodi di I/O di un processo vengono eseguiti altri processi

Lo scopo principale della multiprogrammazione è quello di cercare di sfruttare la meglio le risorse del calcolatore. Questo è particolarmente utile nei sistemi multiutente (per esempio i server); ma anche nei sistemi monoutente esistono innumerevoli attività che possono essere eseguite in modo parallelo, per esempio è possibile utilizzare un riproduttore multimediale mentre si lavora con un editore testi.



contemporaneamente



Si dice che un sistema operativo lavori in **time sharing** quando l'esecuzione del processore viene suddivisa in un certo numero di **quanti temporali**. Allo scadere di un quanto, il processo corrente viene interrotto e l'esecuzione passa ad un altro processo

Il concetto di **time sharing** è l'estensione logica della multiprogrammazione: invece di alternare l'uso del processore ogni qual volta viene eseguita un'operazione di I/O, il sistema operativo passa da un processo all'altro periodicamente, dando a tutti i processi una possibilità di avanzare nel loro programma.

I passaggi avvengono così frequentemente (nell'ordine di centinaia di migliaia di volte al secondo) che i programmi interattivi danno l'impressione di avanzare come se avessero un processore dedicato.

Generazione 0

Primo tentativo di costruzione di una macchina programmabile ad opera di Babbage (1792-1871). Purtroppo la tecnologia dell'epoca non era sufficientemente avanzata per la costruzione di tale macchina.

Generazione 1 (1944 - 1955)

I primi calcolatori programmabili vennero costruiti nel 1944. Queste macchine erano sprovvisti di un sistema operativo, il che significava che poteva essere eseguito un solo programma alla volta.

Generazione 2 (1955-1965)

A partire dal 1955 i nuovi elaboratori sono acquistabili sul mercato. Anche se vengono dotati di S.O (ancora rudimentali) potevo essere eseguiti un processo alla volta, ciò nonostante era possibile inserire più programmi nel calcolatore ed il sistema operativo si occupava di eseguirli uno dopo l'altro

Generazione 3 (1965 – 1980)

A partire dagli anni '60 gli enormi progressi nel campo dei circuiti integrati permisero di costruire calcolatori sempre più potenti. In questo periodo vennero sviluppati due concetti di fondamentale importanza per i sistemi operativi: la **multiprogrammazione** e il **time sharing**

Generazione 4 (1980-)

A partire dagli anni '80, nasce l'ultima generazione dei sistemi operativi utilizzati per controllare i personal computer a basso costo.

Sistemi operativi Microsoft

I sistemi Microsoft sono sicuramente i più diffusi. Esistono due grandi famiglie dei prodotti software nel campo dei sistemi operativi:

1. La prima famiglia comprendeva, Windows 95, Windows 98 e Millenium Edition. Tutti traggono origini dai vecchi sistemi operativi DOS, aggiungendo (anche se in modo incompleto funzioni di time sharing e multiprogrammazioni)
2. La seconda famiglia, che include sistemi quali Windows NT, Windows 2000, Windows XP, Windows 7, è caratterizzata da funzioni moderne di time sharing e multiprogrammazione

Sistemi operativi commerciali basati su UNIX

Unix è sistema operativo sviluppato negli anni '70 ed è oggi presente sotto forma di numerosi sistemi operativi commerciali e non, quali, MAC OS, Solaris SUN

Sistemi operativi open source

Oltre alla versioni commerciali di UNIX, le idee rappresentate da UNIX hanno dato origine a numerosi progetti Open Source, fra cui il più noto è certamente LINUX