

## I SISTEMI OPERATIVI

Insieme di programmi che implementano **funzioni essenziali** per l'uso di un sistema elaboratore.

Le funzioni di un S.O. non sono definibili in modo esaustivo e puntuale così come non è possibile effettuare una **classificazione** basata sulla terminologia corrente e che sia esaustiva e non ambigua.

Il S.O. virtualizza le risorse di una macchina reale e crea una macchina astratta che:

- offre un ambiente di lavoro amichevole per gli utenti finali
- alloca a programmi ed utenti specifici le risorse hardware e software disponibili ottimizzandone l'utilizzo
- controlla l'esecuzione dei programmi ed in particolare l'uso della memoria e dei dispositivi di I/O

### Criteri di classificazione di un S.O.

dal punto di vista dell'utente

- **sistemi dedicati**
- **sistemi a lotti (batch)**
- **sistemi interattivi o conversazionali**
- **sistemi in tempo reale**
- **sistemi transazionali**

### Criteri di classificazione di un S.O.

in base all'organizzazione interna

- **sistemi monoprogrammati**
- **sistemi multiprogrammati**
- **sistemi a partizione di tempo (time sharing)**
- **sistemi ad uso speciale**

## Criteri di classificazione di un S.O.

in base all'architettura del sistema elaboratore

- **S.O. per sistemi di calcolo monolitici**
- **S.O. per sistemi di calcolo distribuiti**
- **S.O. per sistemi di calcolo paralleli**

## Sistemi dedicati

La macchina è dedicata all'uso da parte di un singolo utente.

Il S.O. fornisce un interprete di comandi, la possibilità di lanciare programmi, la gestione di file e dati.

Sono di solito chiamati **Supervisori** per evidenziare che controllano l'esecuzione dei lavori o **DOS** per evidenziare il supporto utilizzato per i file gestiti.

Si caratterizzano per il basso sfruttamento durante le interazioni con l'utente e per l'assenza della virtualizzazione delle risorse.

### **Sistemi a lotti (batch)**

Privilegiano lo sfruttamento del sistema a prezzo dell'assenza di interazione da parte dell'utente che deve fornire in via preliminare il suo lavoro.

I lavori sono raggruppati in lotti e memorizzati su supporto magnetico.

Il S.O. gestisce l'avanzamento dei singoli lavori.

Le unità di ingresso/uscita sono virtualizzate

### **Sistemi a lotti (batch)**

#### **Vantaggi**

La macchina è meglio sfruttata grazie alla non interazione con l'utente. Le operazioni di I/O sono più veloci

#### **Svantaggi**

L'utente può anche attendere ore o giorni per avere i risultati del proprio lavoro.

### Sistemi interattivi o conversazionali

L'utente interagisce con il sistema da terminale durante l'esecuzione del proprio lavoro.

Il sistema assegna le risorse agli utenti per quanti di tempo riducendo drasticamente il tempo di risposta ossia il tempo che intercorre tra l'introduzione di dati da un terminale e l'avvio di una risposta da parte del sistema.

#### **Vantaggi**

L'utente opera come con un sistema dedicato.

#### **Svantaggio**

L'overhead introdotto dal S.O.

### Sistemi transazionali

Trattasi di sistemi interattivi destinati ad eseguire **transazioni** ossia sequenze di operazioni elementari che richiedono l'esecuzione ciascuna di un programma o di una sua parte.

In un sistema transazionale l'utente interroga ed aggiorna archivi.

Tutti gli utenti operano sugli stessi archivi e richiedono l'esecuzione delle stesse operazioni elementari.

Sovente sono anche detti impropriamente sistemi in tempo reale.

## Sistemi in tempo reale

Sono sistemi che gestiscono programmi che interagiscono con l'ambiente esterno attraverso periferiche e che garantiscono a dati di ingresso risposte in un tempo utile rispetto alle costanti di tempo proprie dell'ambiente esterno.

Il tempo utile dipende dall'applicazione e quindi dall'ambiente esterno.

Nell'uso commerciale il termine "real time" è sinonimo di "conversazionale" o "transazionale".

Il vero sistema in tempo reale è un sistema la cui correttezza dipende dalla velocità di esecuzione.

## Le funzioni di un S.O. viste dall'utente

Un utente ha la visibilità delle funzioni di un S.O. tramite i comandi che permettono di richiamarle.

I comandi possono essere:

- **espliciti** (comandi del JCL)
- **impliciti** generati da un compilatore o assembler

I comandi espliciti sono gestiti dall'interprete di JCL

I comandi espliciti e impliciti possono talora attivare le stesse funzioni

Le funzioni di un S.O. viste dall'utente possono raggrupparsi in:

- **funzioni per la gestione dei lavori**
- **supporti per la programmazione**
- **meccanismi di I/O**
- **gestione archivi**

### **Funzioni per la gestione dei lavori**

consentono ad un utente di:

- **qualificarsi nei confronti di un sistema**
- **richiedere l'assegnazione di alcune risorse**
- **richiedere l'attivazione di una o di una successione di azioni**

e di effettuare operazioni accessorie che consentano di addebitare l'uso del sistema

## Supporti per la programmazione

Sono svolte da programmi del S.O. che supportano lo sviluppo software

Tra questi si segnalano:

- **programmi per il trattamento di testi**
- **programmi per l'identificazione di errori**
- **programmi per la gestione di librerie di programmi applicativi e delle relative versioni**
- **programmi di supporto al testing**

## Meccanismi di I/O

Sono funzioni del S.O. che rendono disponibili all'utente meccanismi di I/O potenti ed indipendenti dalle caratteristiche delle periferiche.

Tipicamente consentono di:

- **mascherare conflitti nell'uso di periferiche condivise**
- **mascherare le modalità con cui le operazioni di I/O sono realizzate**
- **gestire entro certi limiti eventuali malfunzionamenti di I/O**

## Funzioni di gestione archivi

Sono funzioni che consentono di manipolare dati su supporti di memorie di massa.

Tipicamente consentono di:

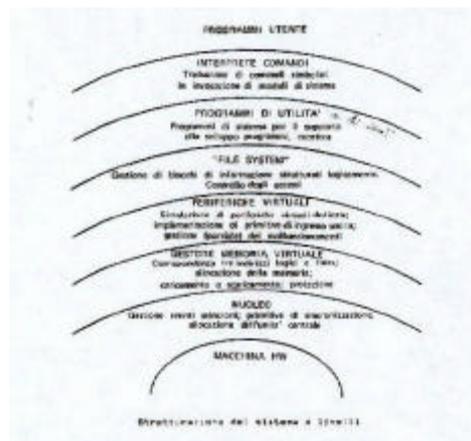
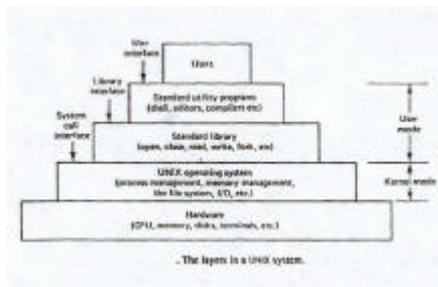
- **accedere con riferimento ad un livello logico**
- **garantire la sicurezza degli accessi**
- **assicurare l'integrità delle informazioni**

## La struttura di un S.O.

Viene solitamente rappresentata con un modello a **livelli di macchine virtuali**.

Ciascun livello:

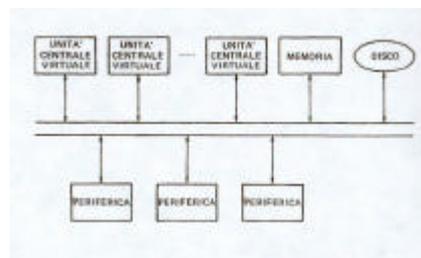
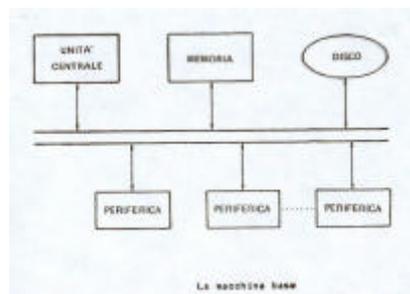
- simula l'esistenza di una macchina virtuale gestendo uno specifico sottoinsieme di risorse del sistema
- è costruito sopra una macchina (virtuale o reale) di livello inferiore utilizzando i servizi che tale macchina offre
- rispetta il principio di separazione fra meccanismi e politiche di gestione
- rispetta i principi della programmazione modulare



## Livello nucleo

Al livello nucleo la macchina virtuale realizzata dal S.O. è una macchina che:

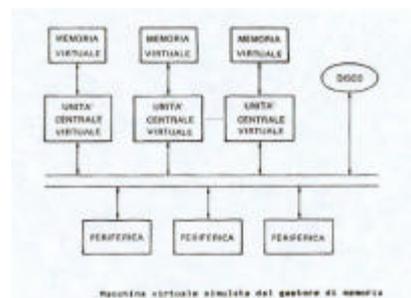
- possiede tante unità centrali quanti sono i processi
- non possiede meccanismi di interruzioni
- possiede istruzioni di sincronizzazione e scambio di messaggio tra processi che operano sulle unità centrali virtuali



## Livello gestione della memoria

Al livello gestione della memoria la macchina virtuale realizzata dal S.O. è una macchina che:

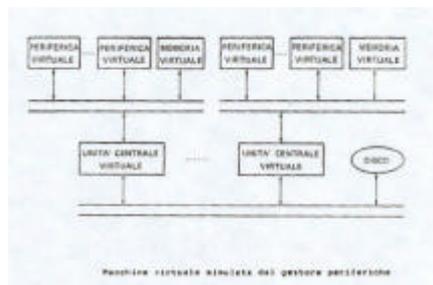
- consente di far riferimento a spazi di indirizzi virtuali
- garantisce la protezione
- consente in taluni casi di ignorare se il programma e/o i dati siano fisicamente residenti in memoria centrale o su memoria di massa (disco)



## Livello gestione periferiche

Al livello gestione periferiche la macchina virtuale realizzata dal S.O. è una macchina che:

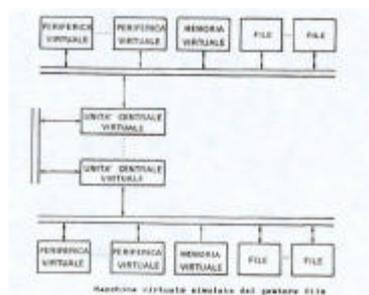
- dispone di periferiche dedicate ai singoli processi
- maschera le caratteristiche fisiche delle periferiche
- gestisce parzialmente i malfunzionamenti delle periferiche



## Livello file system

Al livello file system la macchina virtuale realizzata dal S.O. è una macchina che:

- gestisce blocchi di informazioni su memoria di massa strutturati logicamente
- ne controlla gli accessi
- ne gestisce l'organizzazione



### **Accesso alle funzioni offerte da un livello**

Le funzioni di un livello di macchina virtuale sono rese disponibili ai programmi in esecuzione mediante chiamate di primitive (procedure non interrompibili) del S.O. Il meccanismo base utilizzato per accedere a tali funzioni è quello della SVC e quindi delle interruzioni software generate o mediante specifiche istruzioni assembler o provocando una trap.

Tipicamente le librerie run time dei linguaggi di programmazione rendono disponibili apposite routine di interfaccia che trasformano una tradizionale chiamata di procedura in una SVC con scambio dei parametri tramite registri del processore e/o specifiche aree di memoria

### **Accesso alle funzioni offerte da un livello**

Il meccanismo delle interruzioni è anche utilizzato all'interno dello stesso S.O. per attivare da un livello funzioni del livello inferiore.

Naturalmente il ritorno da un livello inferiore ad uno superiore o al programma che ha eseguito la primitiva avviene mediante una istruzione di ritorno da interruzione

Ad ogni attivazione di una funzione di un livello corrisponde un salvataggio dello stato del processore nello stack di sistema e conseguentemente ad ogni ritorno un ripristino dello stato del processore dallo stack di sistema.