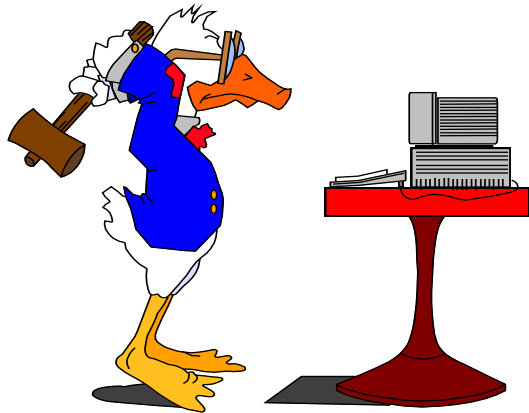


Funzioni di un Calcolatore

- elaborazione dei dati
- memorizzazione dei dati
- trasferimento dei dati
- controllo



- flessibilità nel calcolo
- modularità nell'architettura
- scalabilità dei componenti
- standardizzazione
- abbattimento dei costi

Sistemi operativi

- Gli attuali SO impiegati possono essere visti come una collezione di programmi interagenti che operano sull'hw per fornire agli utenti un insieme di funzioni ad un elevato livello di astrazione
- vantaggi:
 - definire modalità di interfaccia standardizzate verso il calcolatore
 - sviluppare i programmi in maniera più semplice, modulare e indipendente dallo specifico calcolatore
 - permettere l'aggiornamento del sw di base in maniera trasparente

Classificazione dei Sistemi Operativi

■ In base al numero di utenti:

Mono-utente: un solo utente alla volta può utilizzare il sistema

Multi-utente: più utenti contemporaneamente possono interagire con la macchina.

- nel caso di più utenti collegati, il sistema operativo deve fornire a ciascun utente l'astrazione di un sistema “dedicato”

■ In base al numero di programmi in esecuzione:

Mono-programmato: il sistema può gestire l'esecuzione di al più un programma alla volta.

Multi-programmato: il sistema operativo è in grado di portare avanti l'esecuzione di più programmi dando l'impressione della contemporaneità (sebbene ci sia una sola CPU).

- il sistema operativo deve gestire l'unità di elaborazione (CPU) suddividendola tra i vari programmi.

Sicurezza e protezione

■ Controllo degli accessi:

Meccanismi per *l'identificazione degli accessi al sistema*

Procedura di accesso al sistema: *login*

■ A ogni utente è associato uno *username* e una *password*

■ Protezione

Ogni utente può accedere solo a determinati file e risorse

■ Permessi di scrittura, lettura ed esecuzione

Administrator (o *root*) utente privilegiato

■ Accesso a qualsiasi file o risorsa

■ Personalizzazione

Ogni utente può configurare (nei limiti dei permessi ad esso associati) il proprio ambiente operativo

Architettura di un SO

■ Modello a strati

hardware

Nucleo

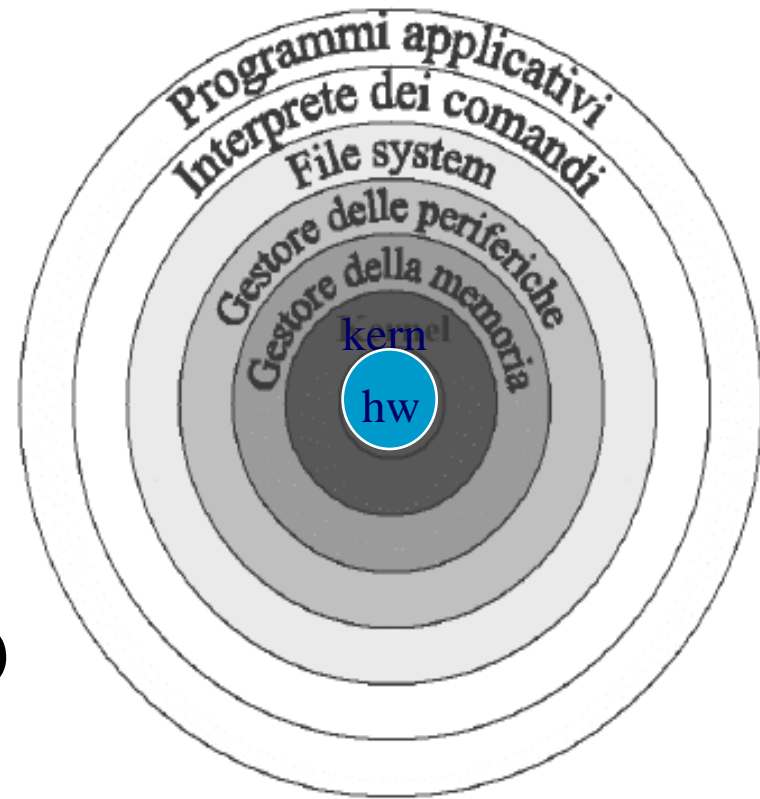
Gestore della memoria

Gestore delle periferiche

Gestore dei file (file system)

Interprete dei comandi (shell)

Applicazioni



Il nucleo del sistema operativo (kernel)

■ Compiti del kernel:

Dialoga direttamente con l'hardware

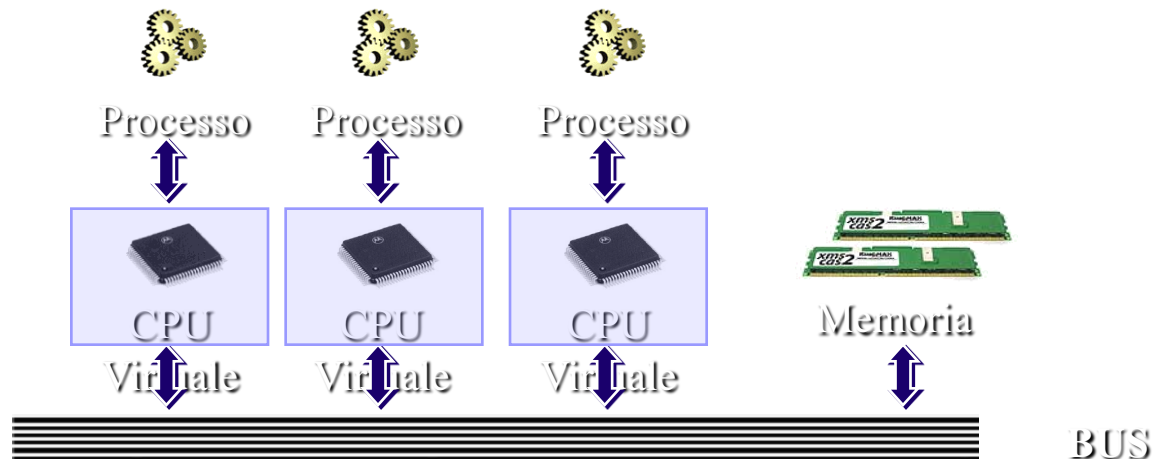
Esegue i programmi utente

Risponde agli eventi (Interrupt) generati dalle periferiche

■ Requisiti Fondamentali:

Consentire a utenti/processi diversi la condivisione delle risorse

Offrire **virtualmente** ad ogni utente/processo una CPU



Programmi e processi

- **Programma: entità statica**

Memorizzato in genere su di un dispositivo di massa

- **Processo: entità dinamica**

Programma in esecuzione

Dati utilizzati dal programma

Informazioni relative al programma (contesto)

- **Ad un programma possono corrispondere diversi processi**

Copie contemporaneamente in esecuzione

- **Un processo può a sua volta richiedere l'esecuzione di altri processi**

processo padre

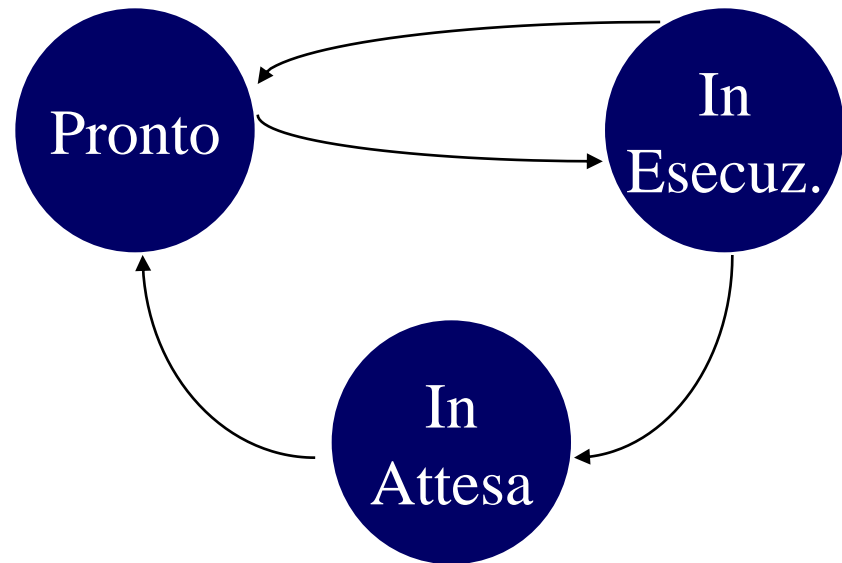
processi figli



Gestione del processore

- Il parallelismo: dati - istruzioni - programmi
- Il processo

“il processo è un programma in esecuzione”



Gli Stati del Processo

- Executing o Running (in esecuzione):
il processore è a disposizione per l'esecuzione del processo. In un calcolatore mono-processore ad ogni istante un solo processo può trovarsi in questo stato
- Ready (pronto):
in grado di essere eseguito ed in attesa del processore dopo il suo rilascio da parte di altri processi
- Waiting (in attesa):
non in grado di essere eseguito perché in attesa di un evento esterno

La multiprogrammazione

- Sistemi uniprogrammati:
quando in memoria centrale, oltre al sistema operativo, risiede un solo processo
- Sistemi multiprogrammati:
quando la memoria centrale può contenere più di un processo

E' il sistema operativo che deve gestire la memoria centrale mantenendo le prestazioni del sistema

Partizione di Tempo

- Sistemi time-sharing:
il tempo viene suddiviso in quanti di tempo da assegnare ai vari processi presenti in memoria
- Preemption e context switch

Lo scheduler

- Lo scheduler è quella parte del sistema operativo che si occupa di gestire i processi. Esso deve:
 - essere efficiente
 - minimizzare il tempo di risposta
 - minimizzare il tempo d'attesa dei processi batch
 - massimizzare il throughput (numero di processi per ora)
 - garantire che ogni processo abbia la sua giusta parte di tempo di CPU

Politiche di scheduling dei processi

■ Round-Robin

Assegnare a rotazione la disponibilità di una unità di tempo (time slice) della CPU ai vari processi

- coda *FIFO* (First In First Out)

Un processo può anche rinunciare al tempo di CPU

- attesa di I/O

■ Round-Robin con priorità

Ad ogni processo viene assegnata una *priorità*

- viene scelto il processo con priorità massima

- I processi ad uguale livello di priorità vengono trattati con politica Round-Robin FIFO

Esempio: Windows 2000

- Round-Robin con priorità (con alcune varianti)

- Quanto: 13-30ms

La concorrenza

- Starvation:
avviene quando più processi richiedono una risorsa ma solo alcuni riescono ad ottenerla
- Deadlock:
avviene quando un gruppo di processi rimane permanentemente bloccato in attesa di una risorsa
- Cooperazione fra processi:
memoria condivisa, scambio di messaggi
- Sezione critica

Il Gestore della memoria

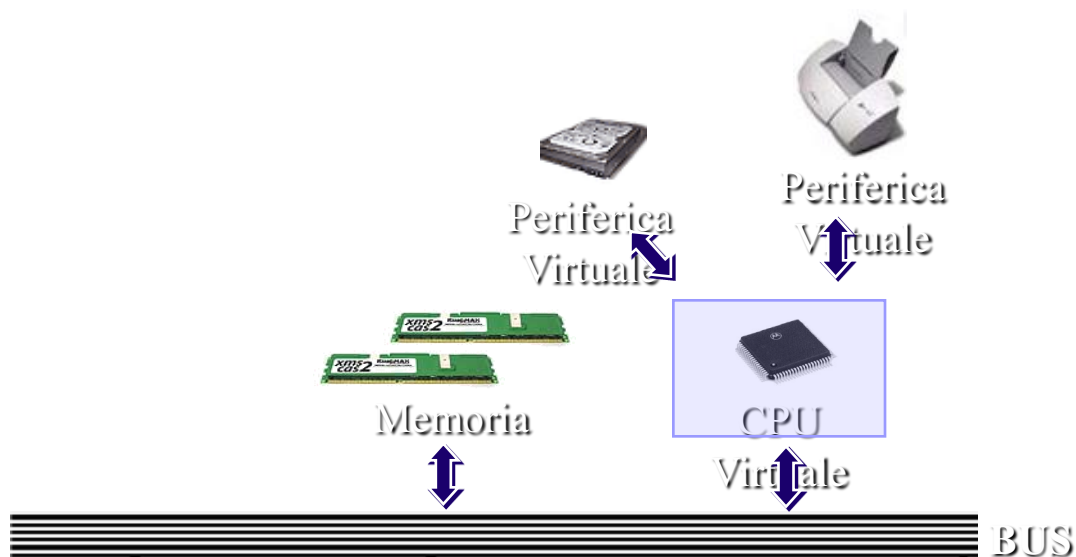
- ha il compito di gestire la memoria in modo trasparente ed efficiente
- Consente ad ogni programma di lavorare in proprio spazio di indirizzamento virtuale
- Deve proteggere i dati e le istruzioni dei programmi
- Deve mascherare la collocazione fisica
- Deve permettere la sovrapposizione degli spazi di memoria associati ai vari programmi

Gestione della Memoria

- Rilocabilità del codice
- Swapping
- Paginazione
- Memoria virtuale
- Segmentazione

Gestore delle periferiche

- Periferiche astratte: Le *caratteristiche fisiche* delle periferiche e le operazioni di I/O che le coinvolgono vengono *mascherate*
- Vengono esposte una serie di *primitive a livello più alto* per leggere e scrivere i dati
- Ogni processo si trova ad operare con periferiche virtualmente dedicate solo ad esso
- *Gestione delle problematiche relative ai conflitti di accesso*



Gestore delle periferiche

■ Drivers

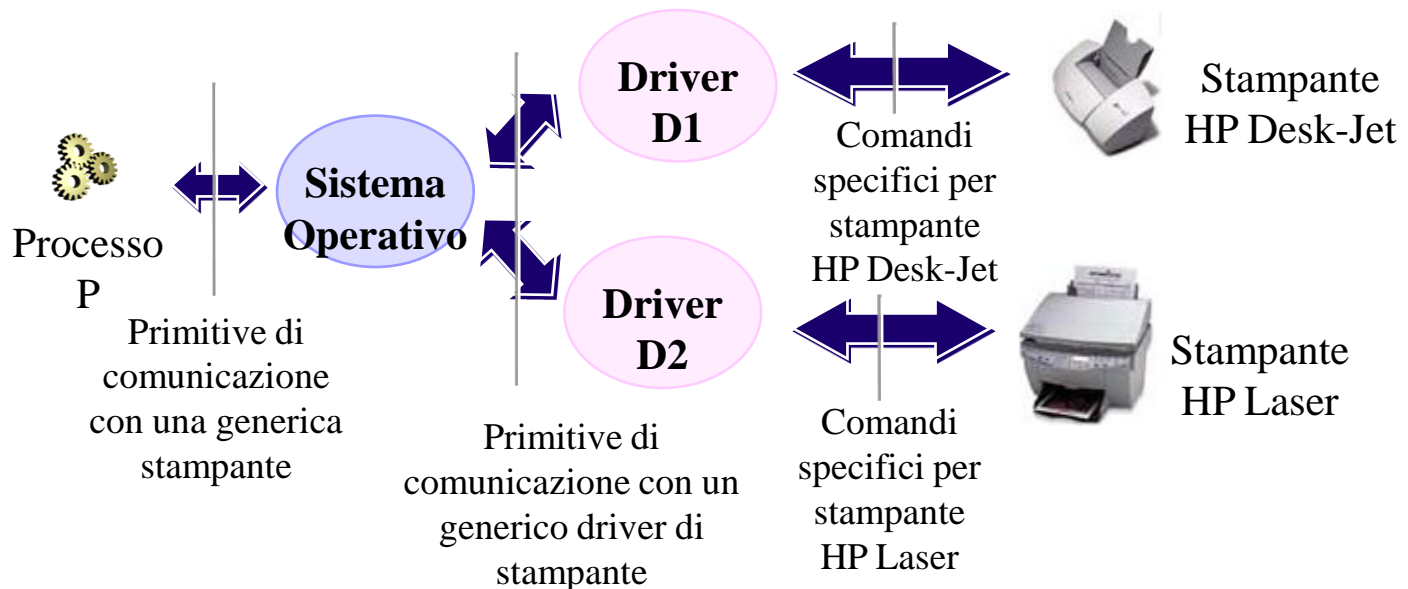
Programmi per la gestione delle periferiche

Inclusi nel sistema operativo

- Spesso sono realizzati e forniti dai produttori delle periferiche stesse

Nascondono al programma applicativo e al resto del SO l'effettiva modalità con cui avviene lo scambio dei dati con le periferiche

- Generalmente tale modalità è diversa per ogni tipo di dispositivo



Gestore dei files (filesystem)

- Funzioni principali di un File System:
 - Fornire un meccanismo di identificazione univoco dei files
 - Implementare i meccanismi per accedere ai files
 - Realizzare metodi per il controllo d'accesso ai files
 - Allocare spazio su disco per la memorizzazione dei files
 - Deallocare e Riutilizzare lo spazio su disco con l'operazione di cancellazione
 - Fornire un'interfaccia utente per: creazione, cancellazione, spostamento, ispezione di file e directory
 - mascherare le caratteristiche fisiche dei dispositivi di memorizzazione e delle interfacce
- I più comuni FileSystems sono:
 - FAT 16
 - Windows3.x, Windows95
 - FAT 32
 - Windows95 (SE), Windows98, Windows ME
 - NTFS (New Technology File System)
 - Windows NT , Windows 2000 Professional e Server
 - NetWare File System
 - Novell Netware
 - Ext2
 - Unix/ Linux

Gestore dei files (filesystem)

■ Il filesystem windows:

Nei Floppy Disk: utilizza il settore come unità minima di allocazione

Negli Hard Disk: non utilizza il settore come unità di allocazione, ma il *cluster*.

■ gruppo di n settori ($n = 2, 4, 8, 16, 32$)

Componenti principali:

■ Partition Table

nel boot sector della partizione stessa e contiene le informazioni sulla partizione.

■ Directory Table

contiene informazioni sui file e le sottodirectory contenute in una directory

■ FAT (File Allocation Table)

permette di individuare i cluster occupati da un file

è il “cuore” del filesystem, per sicurezza essa viene duplicata per proteggerla da cancellazioni o danneggiamenti accidentali

Interazione con l'utente

■ Interprete dei comandi

Riceve i comandi tramite i dispositivi di input

Esegue i programmi associati a tali comandi

- Lettura della memoria di massa del programma da eseguire:

Filesystem

- Allocazione della memoria centrale necessaria e caricamento del programma

Gestore della memoria

- Creazione, attivazione e gestione del processo

Kernel

■ Interfacce utente

A caratteri

Grafiche: GUI (Graphical User Interface)

Ambiente di programmazione

- l'insieme dei programmi che consentono la scrittura e la verifica di nuovi programmi

Editor

- serve per la costruzione di file che contengono testi. In particolare tramite un editor si scrive il *programma sorgente*.

Compilatori

- accettano in ingresso l'intero programma e producono in uscita la rappresentazione dell'intero programma in linguaggio macchina (versione oggetto).

o Interpreti

- traducono ed eseguono direttamente ciascuna istruzione del *programma sorgente*, istruzione per istruzione.

Linker (solo per compilatori)

- nel caso in cui il programma sia suddiviso in moduli (*oggetto*) separati provvede a collegarli per formare un unico *programma eseguibile*.

Debugger

- serve per scoprire ed eliminare errori presenti durante l'esecuzione di un programma, ma non rilevati in fase di compilazione.

Interprete

■ Funzionamento di un interprete:

Preleva un'istruzione I del programma P scritto nel linguaggio L

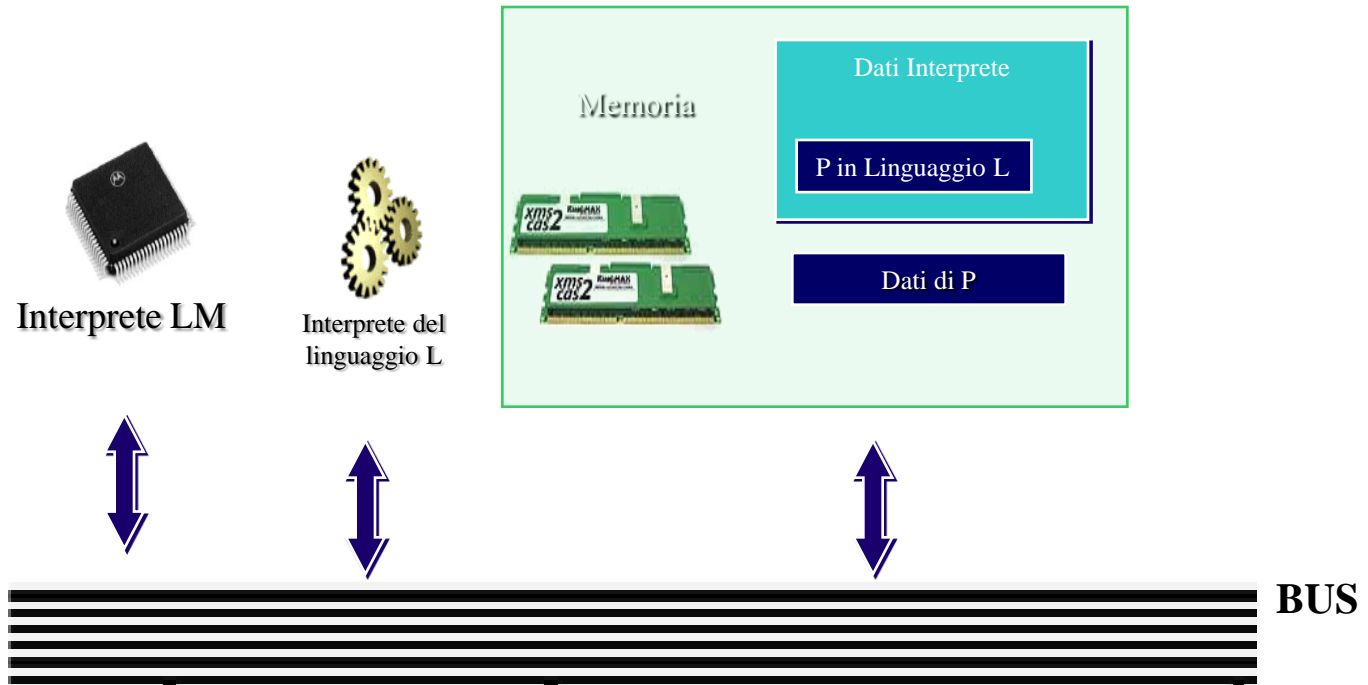
Decodifica I

Traduce I in una serie di istruzioni in linguaggio macchina M_1, M_2, \dots, M_n

Esegue M_1, M_2, \dots, M_n

Passa all'istruzione successiva di P fino a quando non si sia raggiunta una istruzione di terminazione

Interprete



Compilazione e Linking

- È possibile suddividere il programma sorgente in diverse parti

Moduli sorgente

- **Fase di compilazione:**

Ogni modulo sorgente viene compilato producendo l'equivalente in linguaggio macchina

- Modulo oggetto

Eventuali riferimenti a dati o routine di altri moduli vengono raggruppati

- Tabelle dei simboli

- **Fase di linking:**

I moduli oggetto vengono collegati risolvendo i riferimenti contenuti nella tabella dei simboli, producendo un unico programma eseguibile

Compilazione vs Interpretazione

- I programmi commerciali sono solitamente compilati
 - Maggior velocità di esecuzione
 - Protezione del codice sorgente
- Con l'avvento di Internet è stato riavvivato l'interesse per gli interpreti: *JavaScript*, *VBScript*
- Soluzioni miste: Visual Basic, Java, .NET