

SISTEMA OPERATIVO ➤ INTRO

È un'insieme di programmi (software) che gestiscono gli elementi fisici (hardware) del computer.

È allo stesso tempo:

- intermediario tra utente e hardware:

Estende la ISA della macchina, consentendo l'esecuzione di istruzioni complesse in linguaggio macchina, virtualizzandole.

- controllore di programmi utente e hardware:

In particolare:

- Gestisce i programmi in modo da evitare errori e usi scorretti
- Gestisce le risorse hardware quali CPU, GPU, I/O e Memorie

SO - STRUTTURA ➤

L'SO nasce quindi per semplificare l'uso del PC per gli utenti ma anche per i programmi stessi.

I suoi componenti sono:

- Kernel (nucleo):

È la parte principale dell'S.O. e consiste in un programma sempre in funzione **dentro tutti gli altri processi**.

- Programmi di Sistema:

Sono programmi associati al S.O. ma che non è detto che facciano parte del Kernel

- Middleware: (generalmente per mobile)

Software aggiuntivo o supporto di sviluppatori

Il resto sono programmi applicativi.

S.O. - COMMUNIC. CON HW >

L' hardware viene rappresentato come:

- 1 + CPU
- 1 + Device controllers :

Dei controllori dei dispositivi connessi.

Ogni D.o.c. dispone di memoria (buffer) e registri e si contendono l'uso della memoria con la CPU. La quale viene gestita tramite un controllore della memoria.

Essi comunicano tra loro attraverso il Bus.

Il S.O. dispone di 1 driver per ogni device controller, che consente di estrarre la logica della loro gestione

COMUNICAZIONE KERNEL - P. APPLICATIVI >

Può avvenire tramite

- Interrupts : eventi asincroni dei dispositivi
- Exceptions: eventi sincroni da istruzioni.

Interrupts nel dettaglio

La CPU dispone di un pin chiamato INT o interrupt request line che viene controllata dopo ogni esecuzione.

Al rilevamento, la CPU:

1. legge il valore dell'interrupt
2. salva P.C. e registri in registri dedicati.
3. usa il valore come indice nel vettore delle interruzioni.
4. esegue le istruzioni del vettore sovrascrivendo P.C.
5. Ripristina lo stato ed esegue l'istruzione di ritorno (return from interrupt)

FINE LEZIONE 1

Interrupt Chaining

In alcuni casi sono presenti troppi gestori

Per farli entrare nel vettore delle interruzioni

nell'array si fa puntare una collezione di gestori
che verranno invocati uno dopo l'altro fin quando
non si trova quello giusto

Interrupt in sistemi Multi-core

Vengono assegnati ad 1 cpu tramite le logiche
negli interrupt controllers.

Masking e Priorità

Le CPU hanno un sistema di priorità e
due linee di interrupt

- maskable :

Può essere disattivata dalla CPU (es: ci sono
istr. critiche non interruptibili)

- non maskable : Non disattivabile

MEMORIA >

Divisa in:

- volatile: Sovrascrivibile

- non volatile: Per sole letture modificabile ramamente

Interagisce con la CPU tramite load e store.

Viene gestita dall'S.O. tramite la virtualizzazione per:

- Efficientarne l'accesso

- Separare gli spazi di indirizzamento per un processo
per isolarlo.

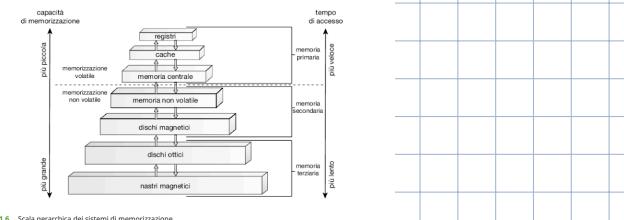


Figura 1.6 Scala gerarchica dei sistemi di memorizzazione.

STRUTTURA DI I/O >

Gli I/O comunicano tramite:

- Interrupt: Per piccole quantità di dati
- DMA

Il device controller sposta blocchi di dati direttamente nella memoria.

La CPU prenderà in carico ogni blocco CPU + interrupt invece di ogni byte, evitando sovraccarichi (overhead).

- MMIO:

Memoria virtuale per trasferire da CPU ad I/O

ARCHITETTURA DEGLI ELABORATORI >

Le architetture attuali sono:

- Sistemi a memoria condivisa (Multi core o Multi proc.)
- Sistemi a mem. NON condivisa (Multicomputer)
- Sistemi con acceleratori HW (GPU, manycore...)
con memoria sia condivisa che non.

Sistemi a mem. condivisa : Dettaglio

Possono essere:

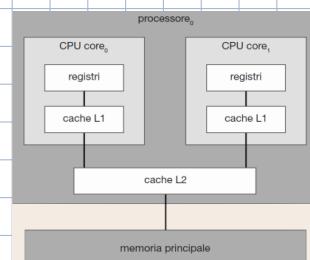
- Ad accesso uniforme a mem.: SMP o UMA

Sistemi in cui vi sono più processori o cores ma che hanno memoria primaria condivisa (insieme eventualmente a cache)

- SMP: Symmetric Multi Processing, ogni proc. può fare tutto.

- Ad accesso non uniforme : NUMA

Ogni CPU o gruppo di CPU ha

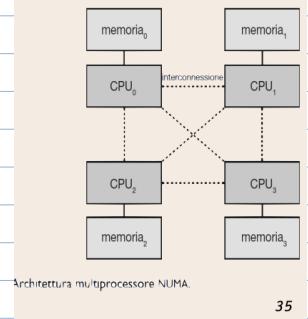


una memoria locale collegata con un mini bus. Perde efficienza nel caso di operazioni cross nella memoria.

L'S.O. gestisce memoria e processi ma può venire modificato dal programmatore

Sistemi a mem. non condivisa

Sono sistemi che condividono mem. di massa e sono collegati con LAN. Ogni Nodo ha un S.O.



ATTIVITÀ DEL SISTEMA OPERATIVO ➤

All'avvio del computer, viene avviato il bootstrap program dal firmware.

Esso deve individuare e caricare il kernel e subito dopo i processi di sistema (o daemon).

Multitasking dei programmi

Vengono caricati più programmi in memoria. La CPU scatta fra i processi

Modalità di esecuzione

• Kernel (0)

Tramite interrupt o eccezioni che usano il vettore delle interruzioni. Ne fanno parte le syscalls

• User (1)

Timer

Timer impostato dall'S.O. per assicurarsi di riacquistare il controllo. Se finisce, viene invocato un interrupt

GESTIONE DEI PROCESSI ➤

Il processo è un'entità attiva. Il programma è entità passiva.

Ogni processo può essere:

- single threaded: 1 prog. counter
- multi threaded: + ↘ ↙

Il sistema sarà quindi l'unione di più processi.

I quali saranno divisi tra p. di sistema e utente

L'S.O. si occuperà di gestire i processi:

GESTIONE DELLA MEMORIA >

Per l'esecuzione, tutte le istruzioni e almeno una parte dei dati deve stare in memoria.

L'S.O. si occuperà quindi di:

- Tenere traccia della memoria in uso
- Decidere quali programmi vanno caricati in mem.
- Assegnare / Revocare gli spazi in memoria.

Inoltre, si occupano di gestire la memoria di massa usando i file e directory per astrarre la logica di gestione della memoria:

- Gestione file / Directory e associazione ad HD.
- Gestione permessi.

PROTEZIONE E SICUREZZA >

Il S.O. si occupa di garantire ai processi:

- Protezione: Del flusso di esecuzione da interferenze non volute
- Sicurezza: Da attacchi e/o pericoli intenzionali interni o esterni.

VIRTUALIZZAZIONE >

Consiste nel dare capacità ai computer di "runnare" più sistemi operativi all'interno di un SO esistente

aggiungendo un gestore delle V.M.

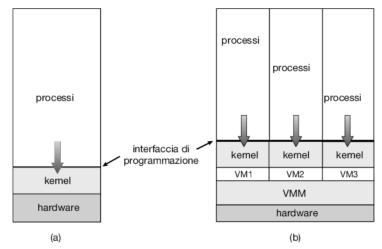


Figura 1.16 Un computer che ha in esecuzione (a) un singolo sistema operativo e (b) tre macchine virtuali.

STRUCTURE DATI >

L'OS usa

- Linked List
- Alberi : Per ricerca
- HashMap: