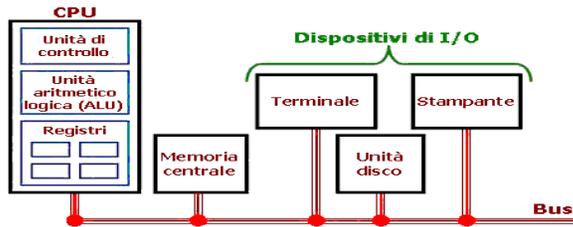


Richiami di concetti del III anno (1)

L'architettura dei moderni computer utilizza un modello ("piantina") del tipo seguente:



Tutte le varie unità di un computer sono collegate tra loro attraverso dei cavi detti BUS

Richiami di concetti del III anno (2)

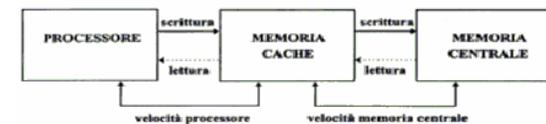
- La memoria centrale è costituita da una sequenza di celle, tutte della stessa lunghezza (o capacità).
- Ogni cella è caratterizzata da un indirizzo che viene utilizzato per leggere / scrivere le informazioni.
- Di solito si sceglie di destinare al nucleo del sistema operativo le prime celle della memoria centrale.

Richiami di concetti del III anno (3)

- La CPU (Central Processing Unit) è il cosiddetto processore di un computer. Essa è costituita da due sottounità principali dette: ALU e CU oltre ad alcuni registri.
- I registri sono celle di memoria interne alla CPU contenenti o dati o indirizzi.
- Esempi di registri sono: MAR e MDR che assieme alla memoria Cache servono a mettere in comunicazione la CPU e la RAM

Richiami di concetti del III anno (4)

- La cache è supporto di memoria che si interpone tra la RAM e la CPU usata per trasferire tra loro piccole quantità di dati in modo veloce.



- Ciascuna CPU ha almeno due livelli di cache L1 e L2 (la cache L2 è più capiente di L1 ma più lenta)
- L'unità disco di figura è rappresentata dai supporti di memoria di massa o ausiliaria alla RAM come: floppy, hard disk, pen drive, dischi ottici, ecc.

Richiami di concetti del III anno (5)

I dispositivi di I/O (di figura) sono:

- **tutte le periferiche di Input** come: tastiera, mouse, scanner, web-cam, penna ottica, joystick, microfono, lettore CD e DVD, schermo tattile, ecc.
- **tutte le periferiche di Output** come: casse, monitor, stampanti, cuffie, videoproiettori, masterizzatori CD e DVD usati solo per scrivere, stampanti, ecc.
- **tutte le periferiche di Input e Output** come: modem, unità drive floppy, masterizzatore CD e DVD usati per leggere e scrivere, ecc

Sistemi Operativi

- **Software**
- Sistema Operativo
- Classificazione dei sistemi operativi
- La macchina virtuale e la struttura del s. o.
- Le funzioni del s.o. più in dettaglio.

Software

Si definisce come **software** l'insieme dei programmi che possono operare sul calcolatore, cioè la componente "leggera" di un elaboratore, in contrapposizione alla parte fisica detta "**hardware**".

Software

Il software di un sistema informatico viene normalmente suddiviso in due categorie:

- **Software di base**
- **Software Applicativo**

Software

- Il **software di base** è dedicato alla gestione delle funzioni elementari dell'elaboratore; tale software lavora direttamente sul livello fisico (hardware) della macchina.
- Esso è costituito dal sistema operativo, da alcuni driver di alcune periferiche e da alcuni programmi di utilità: di svago, di sistema, di testo, utilità per file immagini, musicali e video.

Software

- Il **software Applicativo** è dedicato alla realizzazione di particolari esigenze dell'utente e che riesce ad agire sull'elaboratore solo con il tramite del software di base.
- Esso è costituito ad es. dalla suite office e da tutti i programmi di uso comune dell'utente.

Sistemi Operativi

- Software
- **Sistema Operativo**
- Classificazione dei sistemi operativi
- La macchina virtuale e la struttura del s. o.
- Le funzioni del s.o. più in dettaglio.

Caratteristiche del S. O.

- Il s. o. viene fornito assieme al computer all'atto dell'acquisto.
- E' memorizzato su CD e dopo l'installazione ne esiste una copia sull'hard disk
- Ogni computer può essere usato solo se vi è stato preventivamente installato un s.o.

Esempi di sistemi operativi

- I s. o. più diffusi in ambiente PC sono:
 - Windows: 98SE, 2000Pro, MilleniumE, XP
 - Linux: Red Hat, Suse, Mandrake, Debian, Knoppix
 - MacOSX
- I s. o. più diffusi in ambiente server sono:
Windows NT, 2000 e 2003 server

La release del S.O.

Ogni s.o. è contraddistinto oltre che dal nome da una versione specificata in genere da due numeri o da due parole come:

Windows 98 SE, Linux Mandrake 10.1

ovvero da una sequenza del tipo x.y ove:

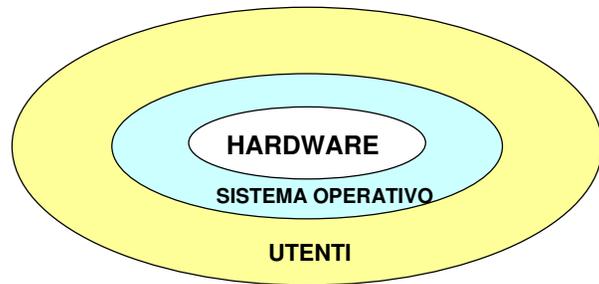
La release del S.O.

- La x rappresenta la vers. principale del s.o. rilasciata per aggiungere nuove funzionalità alla release precedente.
- La y rappresenta il livello di aggiornamento rilasciato per apportare modifiche minori alla release corrente.

Il sistema operativo visto dall'utente



Struttura logica del S.O.



Il sistema operativo agisce da intermediario tra l'hardware e l'utente.

Definizione di S. O.

Più precisamente il s. o. è un insieme di programmi che opera su due livelli:

- gestisce le risorse hardware di un computer come:
 - processore
 - memoria centrale
 - dispositivi di input / output
 - hard disk
 - fornisce l'interfaccia uomo - macchina
 - agisce da intermediario tra la macchina e l'utente
- mascherando all'utente la complessa gestione dell'hw.



Il Sistema Operativo

Il s.o. è suddiviso in due insiemi di programmi:

- un insieme di programmi sempre residenti in memoria centrale (RAM). Esso costituisce il cosiddetto nucleo o nocciolo o kernel del s.o.
- un insieme di programmi che sono caricati in memoria centrale solo quando occorrono.

Operazioni (1)

Alcune operazioni che esso compie sono:

- definire quali programmi sono da eseguire e quali compiti sono da assegnare alla CPU.
- controllare l'allocazione della memoria centrale dei diversi programmi che possono essere "contemporaneamente" in esecuzione.
- garantire l'accesso ai dispositivi di ingresso/uscita

Operazioni (2)

- consentire l'archiviazione e il reperimento dei dati nella memoria di massa.
- permettere agli utenti di accedere in maniera semplice e intuitiva alle funzionalità disponibili.
- supporto per i programmi applicativi.
- controllo degli errori e degli eventi particolari.

Sistemi Operativi

- Software
- Sistema Operativo
- **Classificazione dei sistemi operativi**
- La macchina virtuale e la struttura del s. o.
- Le funzioni del s.o. più in dettaglio.

Classificazione dei S. O.

- **In base all'ambiente in cui operano.**
- In base al modo in cui si presentano all'utente.
- In base all'uso della CPU.
- In base al numero di utenti.

S. O. in base all'ambiente

Tali s. o. si possono ulteriormente suddividere in:

- **s. o. aperti** che sono quelli in grado di "girare" su processori diversi: compatibili IBM e non.
Es. Linux
- **s. o. proprietari** che sono quelli in grado di "girare" solo su un ben determinato tipo di processore.
Es. Windows su compatibili IBM e MacOS su Apple.

Classificazione dei s. operativi

- In base all'ambiente in cui operano.
- **In base al modo in cui si presentano all'utente.**
- In base all'uso della CPU.
- In base al numero di utenti.

S. O. in base al modo in cui si presentano all'utente.

Tali s. o. si possono ulteriormente suddividere in:

- s. o. a batch (o a lotti o a gruppi).
- s. o. interattivi.
- s. o. real time (tempo reale).

S. O. a batch (a lotti o a gruppi)

Essi sono quelli in cui un lotto, cioè un gruppo o un insieme di lavori (o jobs), deciso a priori, viene sottomesso al computer tutto assieme:



che esegue i lavori uno alla volta senza interventi esterni. In essi, quindi, l'utente non interagisce con il computer durante l'esecuzione dei lavori.

S. O. interattivi

Essi sono quelli in cui grazie all'impiego di terminali danno la possibilità all'utente di colloquiare con la l'unità centrale, inviando dati e comandi e ricevendo in breve tempo risposte.

Es. sono quelli usati in ambito gestionale come:

- La gestione di un conto corrente bancario da terminale.
- La prenotazione di un volo o di un treno da terminale.

S. O. real time (in tempo reale)

Essi sono quelli che rispondono “istantaneamente” agli stimoli, alle richieste provenienti dall’ esterno ovvero in tempo utile per influenzare gli ingressi successivi.

Es. sono quelli usati:

- nel controllo e calcolo delle traiettorie aeree
- nel controllo dei processi di produzione di energia nucleare.

Classificazione dei s. operativi

- In base all’ambiente in cui operano.
- In base al modo in cui si presentano all’utente.
- **In base all’uso della CPU.**
- In base al numero di utenti.

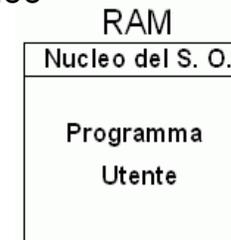
S. O. in base all’uso della CPU

Tali s. o. si possono ulteriormente suddividere in:

- **s. o. in monoprogrammazione.**
- s. o. in multiprogrammazione.
- s. o. in timesharing.

S. O. in monoprogrammazione

Sulle macchine su cui sono installati i s. o. in mono programmazione la memoria centrale (RAM) oltre a contenere il nucleo



contiene un solo programma utente alla volta che deve essere eseguito.

S. O. in monoprogrammazione

Problema:

- Supponiamo che l'unico programma utente in RAM inizi ad usare la CPU.
- Ad un certo istante, prima di terminare, richiede un'operazione di I/O → la CPU in tale situazione cosa fa?

La CPU rimane in attesa che il programma utente termini l'operazione di I/O prima di poter continuare ad eseguire il programma utente.

S. O. in monoprogrammazione

Deduzione:

La CPU che è il componente hw più costoso, rimane in attesa nelle operazioni di I/O quindi non viene continuamente sfruttato.

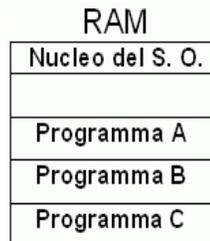
Svantaggio della monoprogrammazione

La CPU è poco utilizzata perché rimane inattiva ogni volta che l'unico programma utente caricato in RAM richiede una operazione di I/O.

Es. S.O. in monoprogrammazione è l'MS-DOS

S. O. in multiprogrammazione (1)

Sulle macchine su cui sono installati i s. o. in multiprogrammazione la memoria centrale RAM oltre a contenere il nucleo



contiene più di un programma utente alla volta.

S. O. in multiprogrammazione (2)

- Pertanto la CPU deve elaborare più programmi ed essa è così veloce che da l'impressione all'utente della contemporaneità di esecuzione. La CPU, però, è una sola e quindi in realtà è sempre un sol programma per volta che viene eseguito.
- Essi sono una alternativa valida ai s. o. in monoprogrammazione perché quando ad es. il programma A richiede una operazione di I/O, la CPU anziché rimanere in attesa, passa ad eseguire il programma B, ecc.

S. O. in multiprogrammazione (3)

Problema:

Supponiamo che il programma A inizia ad essere eseguito dalla CPU ma stavolta esso nè termina e né richiede una operazione di I/O → la CPU non può passare ad eseguire un altro programma.

Svantaggio

In essi, finché un programma in memoria non termina e ne ha richieste di I/O monopolizza la CPU a danno degli altri programmi che rimangono in attesa.

S. O. in timesharing

- Essi nascono dall'esigenza di risolvere il problema dei s. o. in multiprogrammazione, con l'introduzione del concetto di **quanto di tempo**.
- Il quanto di tempo (o time slice o fetta di tempo) è un tempo massimo di utilizzo della CPU scaduto il quale essa passa ad eseguire un altro programma.

Classificazione dei s. operativi

- In base all'ambiente in cui operano.
- In base al modo in cui si presentano all'utente.
- In base all'uso della CPU.
- **In base al numero di utenti.**

S. O. in base al numero di utenti

Tali s. o. si possono ulteriormente suddividere in:

- s. o. **monoutenti (o dedicati)** sono quelli che permettono ad un solo utente alla volta di usare la macchina.
- **s. o. multiutente** sono quelli che permettono a più utenti per volta di usare, di interagire con una macchina.

Sistemi Operativi

- Software
- Sistema Operativo
- Classificazione dei sistemi operativi
- **La macchina virtuale e la struttura del s. o.**
- Le funzioni del s.o. più in dettaglio.

Definizione di Macchina virtuale

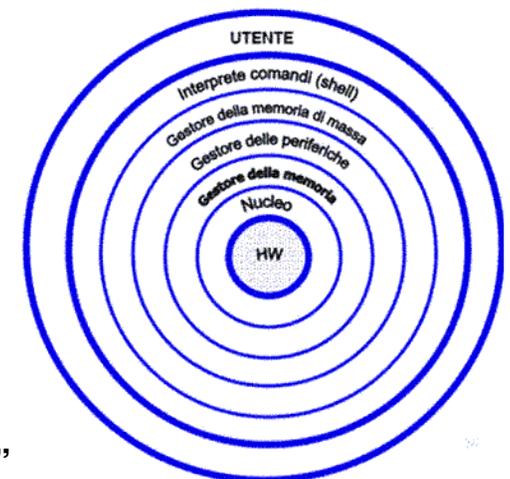
- La macchina virtuale è una macchina che si discosta dalla complessità della macchina reale per sgravare l'utente dalla gestione complessa dell'hardware.
- **Deduzione**
Il sistema operativo + l'hardware della macchina è un esempio di macchina virtuale per l'utente.

Esempio di Macchina virtuale: JVM

- La Java Virtual Machine è un es. di macchina virtuale.
- Essa consiste in un programma che installato nel computer permette di creare una macchina virtuale che si discosta, è indipendente dalle caratteristiche hardware della macchina reale.
- In questo modo le applicazioni java che non sono eseguibili direttamente dalla macchina reale possono essere eseguite dalla macchina virtuale in essa creata più vicina al modo di pensare umano e alle esigenze dell'utente.

Struttura logica a strati (o livelli)

Il s. o. data la sua complessità è suddiviso, per comodità di studio, in cinque strati di programmi detta **struttura "a cipolla"**



I livelli (o strati o moduli) del S. O.

- Ciascun livello offre dei servizi (primitive) ai livelli superiori nascondendo la complessità di ciò che sta sotto.
- **Deduzione**
 - Ciascun livello o strato costituisce una **macchina virtuale** per il livello superiore.
 - La struttura a livelli del s.o. può essere vista anche come una struttura modulare di macchine virtuali.

La struttura modulare delle m. virtuali

- Se ci colloquiamo a livello due del gestore della memoria centrale il livello sottostante uno del nucleo e l'hardware costituiscono una **macchina virtuale del livello uno** in grado di fornire al livello due i servizi richiesti mascherando quello che sta sotto.
- Se ci colloquiamo a livello tre del gestore delle periferiche, il livello sottostante due e la macchina virtuale del livello uno costituiscono una **macchina virtuale del livello due** in grado di fornire al livello tre i servizi richiesti mascherando quello che sta sotto, ecc.

Sistemi Operativi

- Software
- Sistema Operativo
- Classificazione dei sistemi operativi
- La macchina virtuale e la struttura del s. o.
- **Le funzioni del s.o. più in dettaglio.**

Gli strati del sistema operativo

- **Gestore della CPU o Nucleo o Kernel**
- Gestore della memoria centrale
- Gestore delle periferiche o dei dispositivi di I/O
- Gestore della m. di massa o del file system
- Interprete dei comandi.

Che cosa è un processo? (1)

- Per **processo** si intende un programma nel suo divenire (cioè in esecuzione).
- Esso è costituito da una parte **statica** costituita dalle istruzioni del programma e da una parte **dinamica** determinata dai dati di ingresso su cui il programma opera e dal suo stato di avanzamento che indica a che p.to dell'esecuzione esso si trova.

Che cosa è un processo? (2)

- Un esempio di programma è la ricetta della torta alle fragole.
- Un es. di processo è l'esecuzione della ricetta, cioè la creazione della torta alle fragole.
- Tale processo è costituito da una parte statica, la ricetta e da una parte dinamica, gli stati che esso attraversa come: impasto crudo, impasto cotto, torta con crema e torta con crema e fragole.

I processi: istanze di un programma

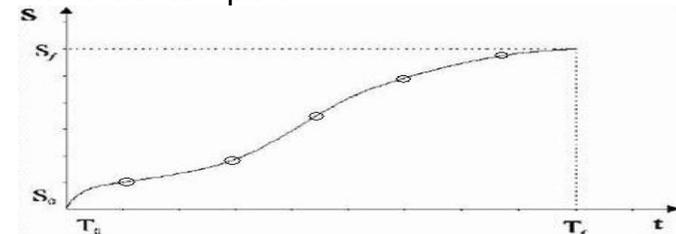
Esempio reale

Se "lanciamo" due volte il programma winword.exe e in uno non scriviamo niente mentre nell'altro scriviamo una frase stiamo generando due istanze o **processi diversi** dello stesso programma winword.exe (usa il task manager di Windows).

Se il numero delle istanze aumentano il programma winword.exe caricato in m.c. è sempre lo stesso mentre (come mostra il task manager di Windows) aumenta lo spazio di m.c. occupato dal programma e dalle relative istanze.

Che cosa è un processo?

Un processo evolve nel tempo, poiché il suo stato S è funzione del tempo t .



Il nostro obiettivo è "fotografare" lo stato di un processo in cinque istanti corrispondenti ai cinque stati principali in cui un processo si può trovare.

Gli stati di un processo (1)

- Supponiamo di usare un sistema operativo in timesharing (la quasi totalità dei s.o. attualmente in uso)
- In memoria centrale, in tale situazione:
 - coesistono diversi processi
 - ciascun processo, come suddetto, si può trovare in uno dei stati seguenti:

Gli stati di un processo (2)

- **new** (o creato) quando il processo viene creato.
- **ready to run** (o pronto) quando il processo è pronto ad usare la CPU non appena arriva il suo turno.

Più processi si possono trovare in tale stato. Per questo motivo essi sono raggruppati in una lista.

- **running** (o esecuzione) quando il processo sta usando la CPU. C'è sempre un solo processo che si può trovare in tale stato poiché la CPU è unica.

Gli stati di un processo (3)

- **wait** (o bloccato o in attesa) quando il processo sta aspettando che termini un'operazione di I/O (es. immissione dati, stampa di dati, ecc).

Più processi si possono trovare in tale stato. Per questo motivo essi sono raggruppati in una lista.

- **terminated** (o end o terminato) quando è stato eseguito completamente dalla CPU.

I passaggi di stato di un processo (1)

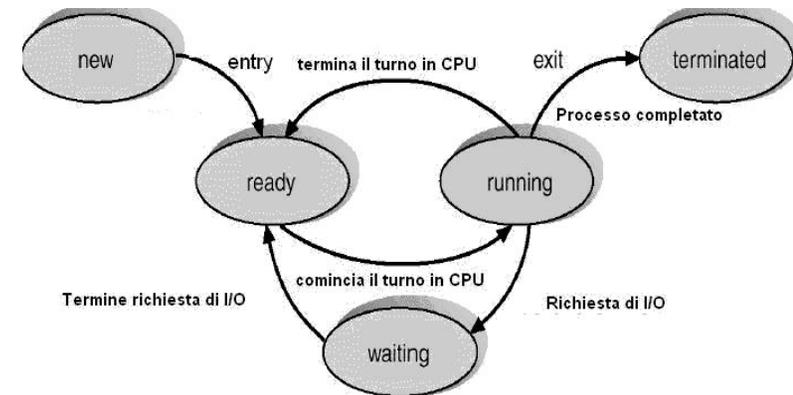
- **da new a ready** quando il processo è pronto ad usare la CPU.
- **da ready a running** quando al processo in testa alla lista dei processi pronti gli viene assegnata la CPU.
- **da running a ready** quando al processo in esecuzione scade il time slice (tempo massimo) di uso della CPU. Esso viene inserito in coda alla lista dei processi pronti.

I passaggi di stato di un processo (2)

- **da running a wait** quando il processo in esecuzione richiede una operazione di I/O
- **da wait a ready** quando il processo ha terminato l'operazione di I/O. Esso viene inserito di nuovo in coda alla lista dei processi pronti.
- **da running a terminated** quando il processo è stato eseguito completamente.

Un processo dallo stato di new passa da uno stato all'altro fino a che non è eseguito completamente.

Il diagramma degli stati di un processo



Esso riassume graficamente i cinque stati principali di un processo con dei cerchi e il passaggio di stato di un processo con delle frecce orientate.

Perché si vuole gestire la CPU?

- La CPU (**C**entral **P**rocessing **U**nit) è una delle unità principali di un computer ed è anche molto costosa e quindi deve essere sfruttata al massimo.
- Poiché la CPU è unica e i processi pronti sono tanti essa deve essere gestita in modo che venga "condivisa" dai "vari" processi in esecuzione.
- La gestione della CPU è affidata ad una parte del s.o. detto appunto **gestore della CPU**.

Gestore della CPU o nucleo o kernel (1)

Il gestore della CPU o nucleo è lo strato del s. o. :

- più vicino all'hardware (livello 1) e quindi è strettamente dipendente da esso.
- sempre residente in memoria centrale, perché il più usato.
- comprende tutti i programmi (routine) di risposta alle interruzioni d'uso della CPU che possono essere interne al processo (richiesta di I/O) o esterne al processo (scade il time slice)

Gestore della CPU o nucleo o kernel (2)

Il gestore della CPU o nucleo è lo strato del s. o. :

- che contiene per ogni processo il **descrittore del processo**.
- che assegna la CPU ai diversi processi pronti secondo particolari regole o norme dette **politiche di scheduling** (gestione) come ad es. : Round Robin, Round Robin con priorità, ecc.

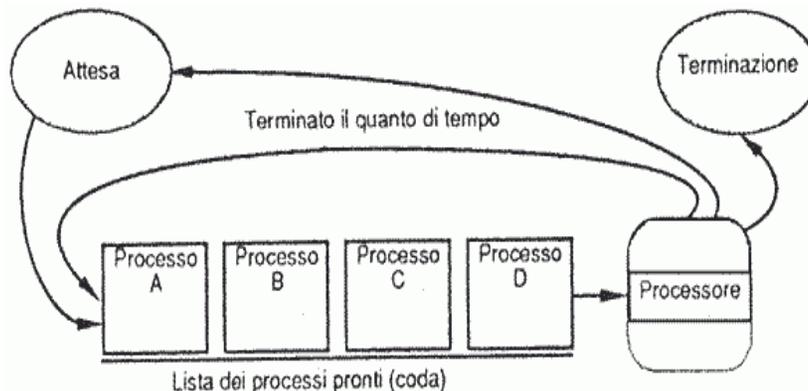
Il descrittore del processo

Il descrittore del processo elenca tutte le informazioni caratterizzanti un processo come:

- stato del processo
- a quale istruzione si è interrotta la sua esecuzione e su quali dati stava operando.
- quante volte ha usato la CPU
- quante operazioni di I/O ha svolto.
- quali pagine del processo sono presenti in memoria centrale (vedi gestore della memoria)

La politica di gestione del R. Robin (1)

- Nella politica di gestione del Round Robin i processi pronti, organizzati in una lista, sono assegnati alla CPU a rotazione tutti per lo stesso time slice.



La politica di gestione del R. Robin (2)

Analizziamo la politica del RR nelle tre situazioni principali seguenti:

1) Quando ad es. un processo A, ha terminato il time slice, se esso non è passato nello stato di end, esso è spostato di nuovo in coda nella lista dei processi pronti per aspettare di nuovo il suo turno in CPU.

Il processo A, sarà rieseguito dalla CPU dopo che i processi, che gli sono davanti, B, C e D hanno terminato il loro time slice.

La politica di gestione del R. Robin (3)

2) Nella lista dei processi pronti di figura, supponiamo che il processo in testa (D) vada in esecuzione, cioè passi da pronto a run. In tali ipotesi, se esso:

- termina prima che finisca il time slice assegnatogli (passando dallo stato di run a quello di terminazione)

- richiede una operazione di I/O (passando dallo stato di run a quello di attesa)

il gestore del processore, in entrambi i casi, assegna la CPU al prossimo processo in lista di attesa ovvero al processo C

La politica di gestione del R. Robin (4)

3) Se un processo, prima che termini il suo time slice, richiede una operazione di I/O esso va in stato di attesa. Quando soddisfa la sua richiesta di I/O è rimesso in coda alla lista come i processi che usano tutto il time slice a loro disposizione (**svantaggio**)

Tale politica, quindi, non assegna nessuna priorità ai processi che non hanno sfruttato a pieno il loro time slice.

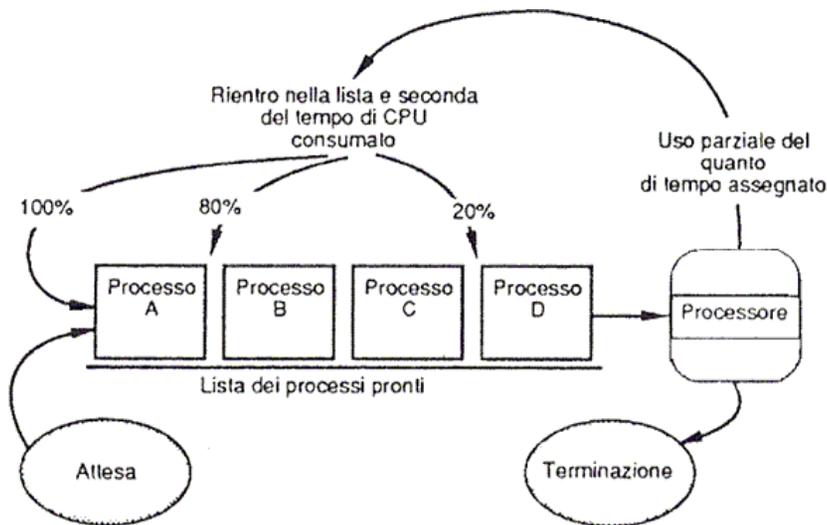
La politica Round Robin con priorità (1)

- E' una evoluzione della politica del Round Robin classica.
- La CPU è assegnata a turno ad ogni processo secondo la tecnica del RR, per il quanto di tempo stabilito.
- I tre casi possibili visti nella politica del RR sono gli stessi anche in questa politica con l'unica differenza di una gestione migliore del 3° caso.

La politica Round Robin con priorità (2)

3) Se il quanto di tempo è usato solo in parte (ad es. per una interruzione di I/O) quando il processo rientra nella lista dei processi pronti la sua posizione verrà determinata in rapporto alla percentuale di tempo di uso della CPU prima della sua richiesta di I/O. Meno tempo di esecuzione usato allora più avanzata sarà la sua posizione nella lista di attesa dei processi pronti.

La politica Round Robin con priorità (3)



© copyright 2005 - IPSIA "A. Parma" - "Il software e il Sistema Operativo" - Prof. M. Simone

69

Gli strati del s. o.

- Gestore dei processi o gestione della CPU o Nucleo o Kernel
- **Gestore della memoria centrale**
- Gestore delle periferiche o dei dispositivi di I/O
- Gestore della m. di massa o del file system
- Interprete dei comandi.

© copyright 2005 - IPSIA "A. Parma" - "Il software e il Sistema Operativo" - Prof. M. Simone

70

Gestione della memoria centrale (1)

- La memoria centrale è una risorsa essenziale per un elaboratore.
- Un programma per essere eseguito deve risiedere all'interno della memoria centrale e lo stesso discorso vale per i dati su cui esso opera.
- Come per la CPU, anche nel caso della memoria centrale esistono dei problemi di gestione come ad esempio:

© copyright 2005 - IPSIA "A. Parma" - "Il software e il Sistema Operativo" - Prof. M. Simone

71

Gestore della memoria centrale (2)

Problema 1

Se ad es. un programma richiede più memoria di quella disponibile, esso non può essere eseguito a meno che non si ricorra a particolari tecniche di suddivisione del programma in parti indipendenti fra loro e quindi singolarmente eseguibili.

© copyright 2005 - IPSIA "A. Parma" - "Il software e il Sistema Operativo" - Prof. M. Simone

72

Gestore della memoria centrale (3)

Problema 2

Se ad es. molti processi devono essere eseguiti *contemporaneamente*, essi dovranno tutti risiedere in memoria centrale perciò si dovrà trovare un meccanismo per permettere a questi processi di condividere l'uso della memoria principale.

Gestore della memoria centrale (4)

Soluzione

Per entrambi i problemi presentati e per tutti gli altri problemi che riguardano la gestione della memoria centrale esistono due tecniche principali:

- **tecniche virtuali** e
- **tecniche non virtuali**

Tecniche non virtuali

Le tecniche non virtuali:

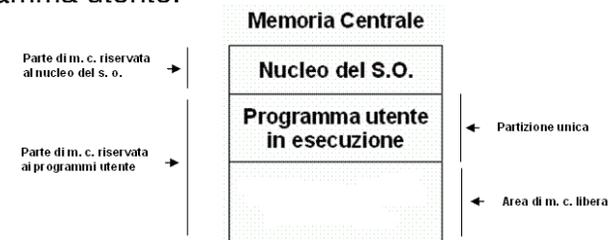
- usano solo la memoria centrale
- *tutto* il processo deve essere caricato in m.c.

Esempi:

- A partizione singola
- A partizioni multiple a dimensione fissa
- A partizioni multiple a dimensione variabile

A partizione singola

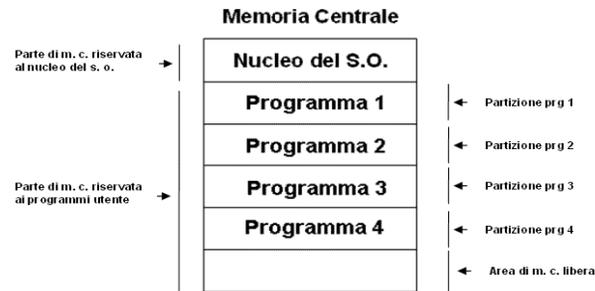
- Essa è la tecnica usata nei s. o. in monoprogammazione. La parte di m. c. riservata ai programmi utente è suddivisa in una sola sezione detta partizione contenente l'unico programma utente.



- **Svantaggio:** quando il programma utente ha dimensione minore rispetto a quella disponibile, una parte della m. c. rimane libera (inutilizzata) allora c'è uno spreco di m. c.

A partizioni multiple

- Essa è la tecnica usata nei s. o. in multiprogrammazione.



- La parte di m.c. riservata ai programmi utente è suddivisa in più sezioni dette partizioni contenente ognuna uno solo dei più programmi contemporaneamente residenti in m.c.

- Tale tecnica può essere a dimensione fissa o a dimensione variabile

A partizioni multiple a dimensione fissa (1)

- In tale tecnica la dimensione delle diverse partizioni deve essere fissata a priori e non può essere in seguito modificata.

Svantaggi

- Il numero massimo di partizioni fisse in cui la m. c. è suddivisa è prefissato quindi l'ho è anche il numero di programmi che si possono caricare in m. centrale.

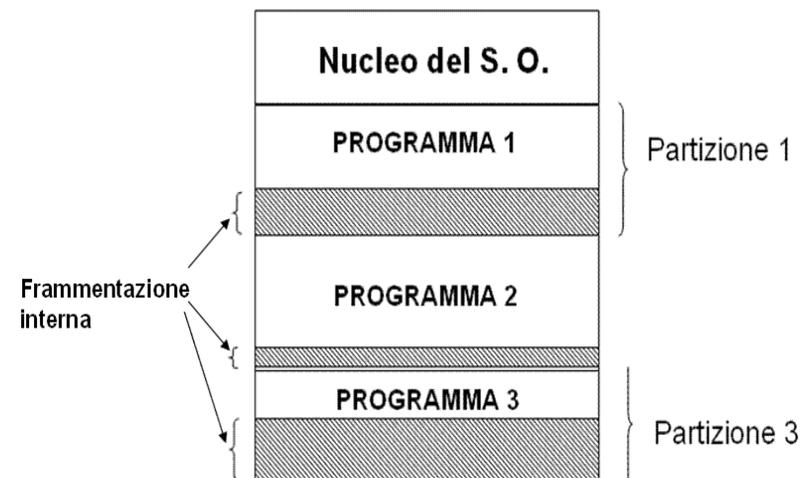
A partizioni multiple a dimensione fissa (2)

Svantaggi

- Partizioni troppo piccole renderebbero impossibile il caricamento in m.c. di programmi ampi
- Partizioni troppo grandi provocherebbero spreco di memoria centrale perché i programmi occupano mediamente solo una parte della partizione.

Esso genera il fenomeno della **frammentazione interna** (vedi fig.)

A partizioni multiple a dimensione fissa (3)



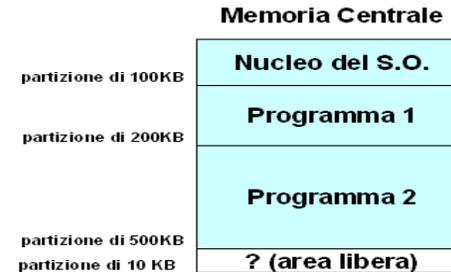
A partizioni multiple a dimensione fissa (3)

Possibili migliorie:

- Definire partizioni di diverse dimensioni: piccole, medie e grandi in modo che un programma “grosso” sarà caricato in una partizione grande mentre i programmi più piccoli saranno caricati nelle partizioni più piccole.
- Tecnica a partizioni multiple a dimensione variabile (o dinamica)

A partizioni multiple a dimensione variabile (1)

- In tale tecnica ciascuna partizione ha dimensione uguale a quella del programma utente da caricare.



Vantaggio: essa, come si nota dalla figura, non genera il fenomeno della frammentazione interna

A partizioni multiple a dimensione variabile (2)

Svantaggi:

- 1) Essa, come si nota dalla figura, genera il fenomeno della **frammentazione esterna**, cioè la m.c. presenta un'area di m.c. libera non utilizzabile da un programma residente in m. di massa di dimensione maggiore dell'area libera in m.c.

A partizioni multiple a dimensione variabile (3)

Svantaggi:

- 2) Se la dimensione del programma utente è piuttosto ampia (ad es. videogiochi) o addirittura più grande dello spazio della m.c. il suo caricamento in m.c. non è possibile.

Possibile miglioria

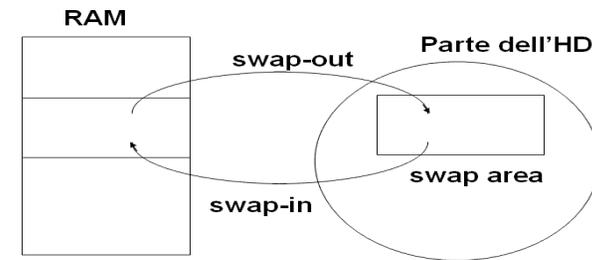
Bisogna virtualizzare le memoria centrale.

Che significa virtualizzare la m.c? (1)

- Significa simulare via software l'esistenza di una m. centrale pressoché infinità ovvero maggiore di quella effettivamente installata.
- In questo modo il programmatore di software è libero di scrivere programmi di dimensioni anche più grandi della capacità della m. c. installata nel computer.

Che significa virtualizzare la m.c? (2)

- La virtualizzazione della m.c. si realizza in pratica:
- aggiungendo alla m.c. effettivamente installata un'area dell' HD detta **area o file di swap o m. virtuale**



- gestendo la m.c. con una tecnica virtuale.

La memoria virtuale

- La m. virtuale è usata come espansione della m. centrale ovvero come deposito temporaneo di informazioni contenute nella m.c. ma attualmente non utilizzate.
- Essa non è accessibile da parte dei programmi utente.
- Il gestore della m.c. sceglie quali parti della m. c scaricare temporaneamente nella m. virtuale e quali parti della m. virtuale riportare in m. c.

Le tecniche virtuali usate

- Tutte le tecniche virtuali usate hanno lo stesso obiettivo: caricare in m.c., solo un "pezzo" di programma per volta lasciando in m. virtuale "parcheeggiati" temporaneamente gli altri "pezzi".
- Quando la CPU richiede ulteriori istruzioni viene caricata un altro "pezzo" del programma e così via fino a eseguire totalmente il programma stesso.

Tecniche virtuali

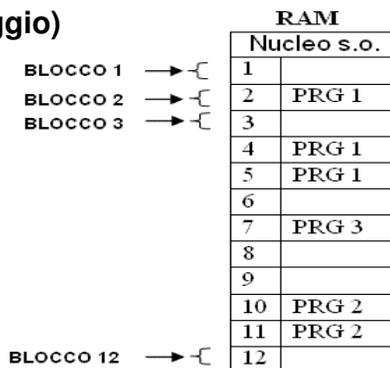
- Tali tecniche di gestione della m.c. usate per la virtualizzazione della m.c. si definiscono tecniche virtuali.
- Esse si classificano in:
 - Paginazione
 - Segmentazione
 - Segmentazione paginata
- Noi ci limiteremo a studiare la **paginazione**.

Paginazione (1)

- Ogni programma è suddiviso in parti sequenziali (una dopo l'altra) di uguali dimensioni dette **pagine**.
- In questo modo per risparmiare spazio in m.c. è possibile mantenere in m.c. solo le pagine che la CPU dovrà eseguire nell'immediato. (**1° vantaggio**)
- La m.c. è suddivisa in parti di dimensioni uguali a quelle delle pagine, detti **blocchi** o frames (in genere di dimensione pari a multipli di 512byte).

Paginazione (2)

Le pagine sequenziali dei programmi utente (PRG1, PRG2, PRG3, ecc) sono memorizzate in blocchi della m.c. non necessariamente sequenziali. (**2° vantaggio**)



Paginazione (3) - tabella delle pagine

Il gestore della m.c. fa uso di una **tabella delle pagine** che mette in corrispondenza le pagine di ciascun programma utente con i blocchi della m.c. in cui esse sono memorizzate.

Tale tabella mostra, in modo particolare:

- In quali blocchi di m.c. sono caricate le pagine di ciascun programma utente.
- Lo stato di ciascun blocco di m.c. e quindi la lista dei blocchi di m. c. liberi.

Paginazione (4)

Un esempio di tabella delle pagine è la seguente:

RAM	
Nucleo S.O.	
1	
2	PGR 1
3	
4	PGR 1
5	PGR 1
6	
7	PGR 3
8	
9	
10	PGR 2
11	PGR 2
12	

Blocco occupato → { 2, 4, 5, 7, 10, 11 }

Blocco libero → { 1, 3, 6, 8, 9, 12 }

Tabella delle pagine			
PGR-ID	N. Pagina	N. Blocco	Stato
		1	libero
PRG 1	0	2	occupato
		3	libero
PRG 1	1	4	occupato
PRG 1	2	5	occupato
		6	libero
PGR 3	0	7	occupato
		8	libero
		9	libero
PGR 2	0	10	occupato
PGR 2	1	11	occupato
		12	libero

© copyright 2005 - IPSIA "A. Parma" - "Il software e il Sistema Operativo" - Prof. M. Simone

33

Paginazione (5) - caratteristiche

- Se la prossima istruzione da eseguire, del programma in esecuzione, non è presente nella pagina del programma caricata in m.c. il gestore della m.c. provvede a caricarla in un blocco di m.c. libero.
- Se tutti i blocchi di m.c. sono occupati il gestore della m.c. provvede a scaricare nella memoria virtuale una delle pagine presenti in m.c.
- Per migliorare l'efficienza della paginazione è scaricata nella m. virtuale la pagina usata meno di recente perché è quella che ha più bassa probabilità di essere usata al momento.

© copyright 2005 - IPSIA "A. Parma" - "Il software e il Sistema Operativo" - Prof. M. Simone

94

Applicazione (1)

La **memoria virtuale** è gestita in modo diverso dai diversi sistemi operativi come:

- Windows 98 / 98SE;
- Windows 2000 / XP;
- Linux nelle varie distribuzioni

Applicazione (2)

- **In Windows 98 / 98 SE** la memoria virtuale è grande, per default, almeno quanto lo spazio libero sull'hard disk ove è installato il s.o.
- Vai in Risorse del computer → proprietà → Prestazione → Memoria Virtuale → Impostazione della memoria virtuale

© copyright 2005 - IPSIA "A. Parma" - "Il software e il Sistema Operativo" - Prof. M. Simone

95

© copyright 2005 - IPSIA "A. Parma" - "Il software e il Sistema Operativo" - Prof. M. Simone

96

Applicazione (3)

- **In Windows 2000 / XP** la memoria virtuale consiste in un file dell'hard disk ove è installato il s.o. detto di **paging** che ha per default la dimensione minima di 288MB che può essere aumentata sino al massimo a 576MB
- Vai in Risorse del computer → Proprietà → Avanzate → Opzioni Prestazioni → Cambia

Applicazione (4)

- **In** qualunque distribuzione di **Linux** la memoria virtuale è una partizione dell' hard disk detta **swap** indipendente dalla partizione dell'hard disk ove è installato il s.o.
- La sua dimensione può essere scelta dall'utente (in genere si sceglie grande quanto il doppio della RAM effettivamente installata) o dalla wizard (procedura di installazione guidata).

Gli strati del s. o.

- Gestore della CPU o Nucleo o Kernel
- Gestore della memoria centrale
- **Gestore delle periferiche o dei dispositivi di I/O**
- **Gestore della m. di massa o del file system**
- **Interprete dei comandi.**

Gestore delle periferiche

- Ha il compito di controllare il funzionamento corretto delle periferiche collegate alla CPU e di renderle disponibili alle applicazioni che le richiedano.
- Possiede gli strumenti per l'aggiunta / rimozione di nuovi driver (ad es. wizard, ecc)
- I driver sono programmi usati per la gestione delle periferiche quasi tutti inclusi nel s. o. e spesso realizzati e forniti dai produttori delle periferiche stesse.

Gestore della m. di massa o del file system (1)

- Esso è lo strato del s.o. che provvede alla gestione delle informazioni residenti sulla m. di massa (hard disk, floppy, dischi ottici, ecc)
- Le informazioni sono memorizzate in modo permanente in m. di massa in “contenitori” detti **file**.
- A ciascun file è associato un **descrittore del file**.

Gestore della m. di massa o del file system (2)

Il **descrittore del file** elenca tutte le caratteristiche di un file come: il nome, il suo formato, la dimensione in byte, la data e l'ora della creazione o dell'ultima modifica, ecc.

I file e i relativi descrittori sono contenuti in dei contenitori “logici” detti **cartelle** o **tabelle dei descrittori**.

Una cartella, quindi, raccoglie, sia i file e sia tutte le sue caratteristiche oltre a contenere altre sottocartelle.

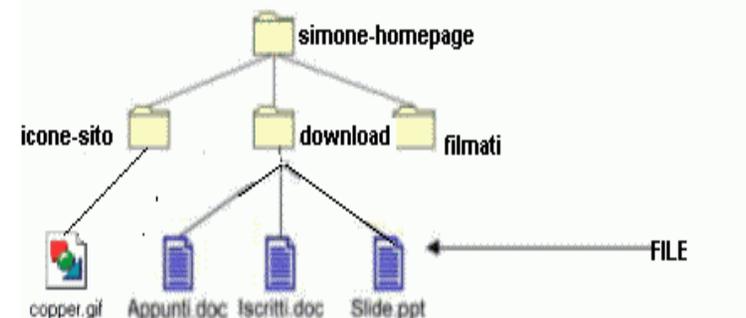
Es. del contenuto di una cartella in Windows

Directory di C:\Documents and Settings\Utente

```
12/12/2005 18.07 <DIR>
12/12/2005 18.07 <DIR>
09/12/2005 23.00          0 AdobeWeb.log
12/12/2005 16.53 <DIR>          Desktop
06/12/2005 23.00 <DIR>          Documenti
25/05/2003 13.52      12.688.384 FaiDaTePro.exe
22/11/2005 00.12      697.567 internet-verri.PDF
30/11/2005 22.06 <DIR>          Menu Avvio
12/12/2005 16.53      167.936 MODULO10.doc
30/11/2005 12.44 <DIR>          Preferiti
01/12/2005 08.06      402.830 ProductContext1310.log
          5 File      13.956.717 byte
          6 Directory 77.913.387.000 byte disponibili
```

Gestore della m. di massa o del file system (2)

L'insieme dei file e delle cartelle memorizzati in una m. di massa sono organizzati secondo una struttura ad albero detta **file system**



Interprete dei comandi (1)

- Esso è l'interfaccia del sistema operativo verso l'utente detta perciò anche **interfaccia utente**.
- Essa riceve i comandi dell'utente tramite i dispositivi di input (mouse, tastiera, ecc) ed esegue i programmi associati.
- L'interprete dei comandi può essere:
 - a caratteri (shell del Dos o di Unix)
 - grafica, cioè con finestre ed icone (Windows)

Interprete dei comandi (2)

- Nei s. o. con un **interfaccia testuale**, i comandi sono forniti dall'utente tramite *stringhe di caratteri*.
- Nei s.o. con una **interfaccia grafica** (GUI) i comandi sono forniti dall'utente tramite la selezione dell'icona che è legata alla operazione richiesta.

Interprete dei comandi (3)

- I vantaggi della GUI sono la **facilità d'uso**, la totale **indipendenza** dall'hardware sottostante.
- Gli svantaggi della GUI sono la lentezza e la grandezza in termini di byte.

Interprete dei comandi (4)

- I vantaggi di una interfaccia a caratteri sono la **poca m. centrale** richiesta e la **velocità operativa elevata**.
- Gli svantaggi di una interfaccia a caratteri sono relativi alla difficoltà di ricordarsi i comandi correttamente (uso di un manuale).