

## Sistemi Operativi – primo modulo

### Introduzione

Augusto Celentano  
Università Ca' Foscari Venezia  
Corso di Laurea in Informatica



## La gestione di un calcolatore

- Un calcolatore (sistema di elaborazione) è un sistema complesso e la sua gestione non può essere distribuita e replicata in tutti i programmi applicativi
  - operazioni ripetitive complesse svolte da tutti i programmi
  - relazioni e interferenze tra programmi diversi
  - controllo del funzionamento del calcolatore come macchina
- Il software di un sistema di elaborazione si può dividere in due classi:
  - *software applicativo*, composto da programmi e servizi che risolvono problemi per gli utenti
  - *software di sistema*, composto da programmi e servizi che gestiscono il funzionamento del calcolatore (del sistema di calcolo)

© Augusto Celentano, Sistemi Operativi – Introduzione



## Sistema operativo (1)

### (Perhaps) Surprising places to find an OS:

Personal digital assistants  
Cable TV controller boxes  
Electronic games  
Copiers  
Fax machines  
Remote controls  
Cellular telephones  
Automobile engines  
Digital cameras

(Elmasri et al., 2009)



© Augusto Celentano, Sistemi Operativi – Introduzione

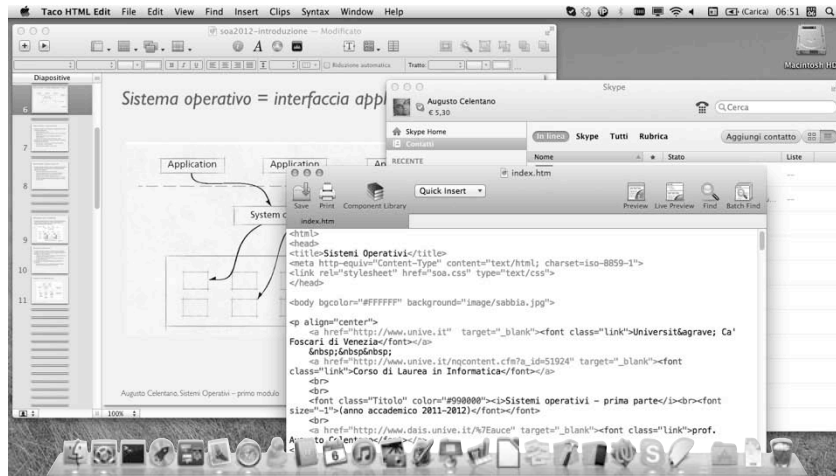
## Sistema operativo (2)

- Il sistema operativo può essere visto e studiato da diversi punti di vista:
  - come *interfaccia tra l'utente e il calcolatore* permette di utilizzare il sistema di calcolo e le sue risorse per risolvere problemi
  - come *interfaccia tra le applicazioni e il sistema* permette al software applicativo di usare in modo controllato le risorse della macchina
  - come *macchina virtuale* permette di programmare come se si avesse a disposizione una macchina funzionalmente estesa e protetta
  - come *gestore di risorse* controlla e coordina il funzionamento contemporaneo dei componenti del sistema
- I diversi punti di vista non si escludono a vicenda
  - in un sistema operativo coesistono servizi per l'esecuzione dei programmi, funzionalità per la gestione interna e funzioni standard per la programmazione

© Augusto Celentano, Sistemi Operativi – Introduzione



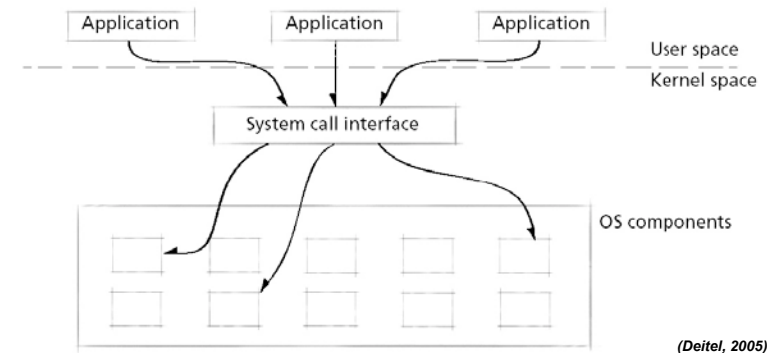
## Sistema operativo = interfaccia utente - sistema



© Augusto Celentano, Sistemi Operativi – Introduzione

5

## Sistema operativo = interfaccia applicazioni - sistema



© Augusto Celentano, Sistemi Operativi – Introduzione

6

## Sistema operativo = macchina virtuale

- Il sistema operativo racchiude funzioni che gestiscono in modo standard situazioni quali
  - operazioni di ingresso e uscita
  - presenza contemporanea di più programmi in memoria che si alternano nell'esecuzione
  - risposta ad eventi esterni (tempo, segnalazioni dalla periferia, malfunzionamenti)
  - adattamento alla varietà dei dispositivi di memoria e esterni
- Dà agli utenti la visibilità di una macchina estesa più semplice da programmare e più protetta
  - tanti programmi → tanti processori, tante memorie indipendenti
  - tante periferiche → gestione unificata
  - strutture di archivio complesse → visione logica

© Augusto Celentano, Sistemi Operativi – Introduzione

7

## Sistema operativo = gestore di risorse

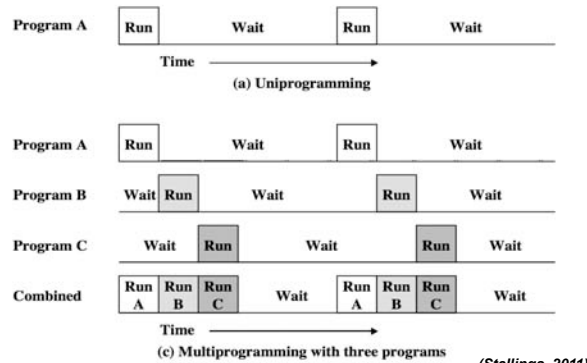
- Il sistema operativo gestisce le risorse del sistema (componenti, sottosistemi, tempo di elaborazione, etc.) distribuendole alle attività in corso (*processi*)
  - utilizza l'unità centrale a turno per i diversi processi
  - conserva più programmi e dati in memoria evitando interferenze
  - sincronizza le attività comuni e l'uso di informazioni condivise
  - stabilisce le priorità di intervento necessarie nei vari casi
  - protegge le informazioni private degli utenti da accessi non autorizzati
  - simula per ogni utente un sistema di elaborazione dedicato e completo (*macchina virtuale*)

© Augusto Celentano, Sistemi Operativi – Introduzione

8

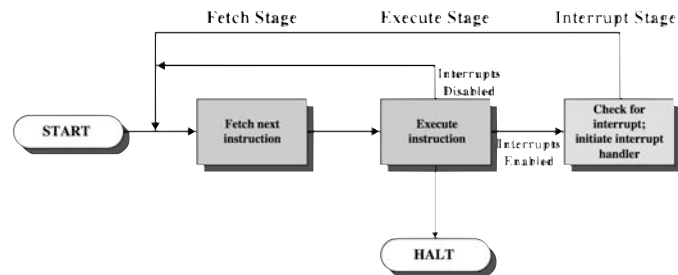
## Multiprogrammazione (multitasking)

- Il modello di elaborazione più comune si basa sull'alternanza di momenti di puro calcolo con momenti di interazione con l'esterno del sistema CPU-memoria



**(Stallings, 2011)**

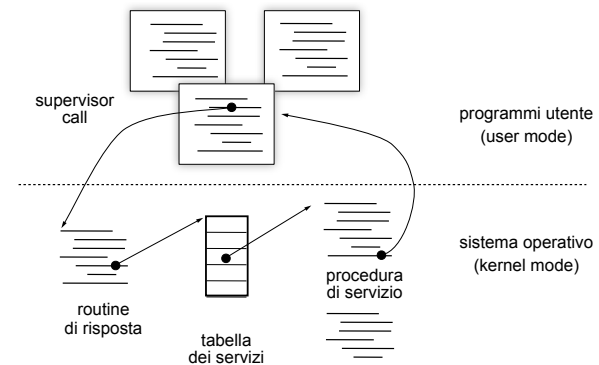
## Gestione delle interruzioni



(Stallings, 2011)

## Interruzioni e supervisor call

- Il sistema operativo interviene su richiesta di un programma (*processo*) o in seguito ad un evento che modifica o richiede di modificare lo stato del sistema
  - la richiesta di intervento da parte di un processo avviene attraverso una *chiamata al supervisore (supervisor call, SVC)*
  - la richiesta di intervento a seguito di un evento avviene attraverso il meccanismo delle *interruzioni*
- In entrambi i casi vengono attivati gli stessi meccanismi di esecuzione
  - una *interruzione software* è una richiesta di intervento del sistema operativo non causata da dispositivi esterni alla CPU

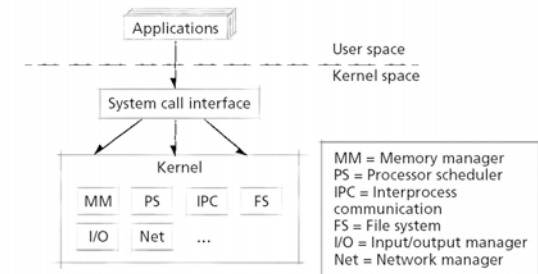


## Classificazione strutturale dei sistemi operativi

- Sistemi monolitici
- Sistemi a livelli (*layered*)
- Sistemi a microkernel
- Sono modelli di riferimento che a volte contemplano soluzioni intermedie che evolvono nel tempo
  - es. sistemi monolitici con moduli dinamici

## Sistema operativo monolitico (I)

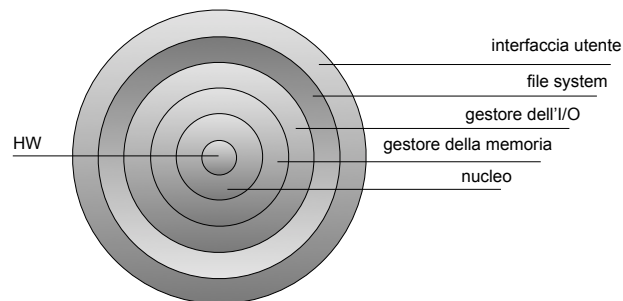
- I primi sistemi presentavano un'interfaccia unica e complessiva verso la macchina e le sue risorse
  - complessità di gestione delle relazioni tra le diverse funzioni
  - scarsa modificabilità
  - ingestibile per sistemi complessi



(Deitel, 2005)

## Sistema operativo a livelli (*layered*)

- I sistemi operativi moderni definiscono una serie di *macchine virtuali*, realizzate per mezzo di *astrazioni* sopra la macchina fisica
  - è uno schema di riferimento di massima largamente adottato



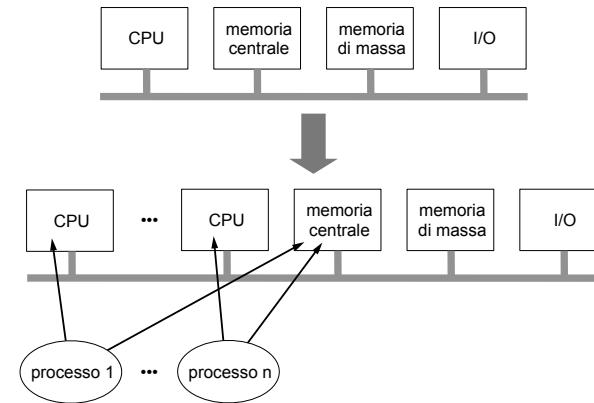
## Nucleo di un sistema operativo (I)

- In un sistema ideale dedicato alla esecuzione di un solo programma le operazioni sono sequenziali e sincrone
  - tutte le operazioni (elaborazione, ingresso e uscita, controllo dell'esecuzione) sono eseguite una dopo l'altra in modo deterministico e ripetibile
  - ogni azione viene terminata prima di passare all'azione successiva
  - l'esito dell'elaborazione non dipende dal tempo totale di esecuzione, né dal tempo relativo di esecuzione delle singole operazioni
- In un sistema reale multiprogrammato tale comportamento può essere riferito al singolo programma, ma non al sistema nel suo complesso

## Nucleo di un sistema operativo (2)

- La macchina è dotata di un solo processore e di una sola memoria centrale. Il nucleo ripartisce l'uso della unità centrale tra i diversi processi attraverso la gestione delle interruzioni
  - si genera una macchina virtuale in cui ad ogni programma attivo (processo) corrisponde una unità centrale virtuale dedicata.
  - i programmi che vengono eseguiti in tale ambiente non devono occuparsi della ripartizione dell'uso dell'unità centrale, poiché ciascuno ne utilizza una diversa (virtuale)
- Alla base di questo livello di virtualizzazione c'è il meccanismo degli *interrupt* e delle *supervisor call*

## Nucleo di un sistema operativo (3)

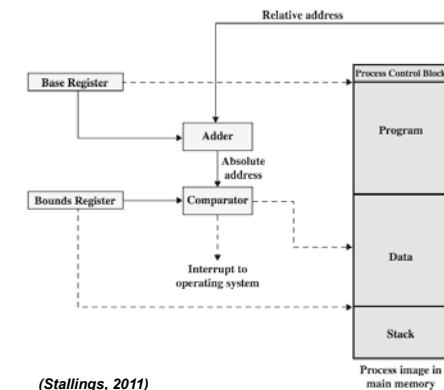


## Gestore della memoria centrale (1)

- Il gestore della memoria di un sistema operativo consente di programmare riferendosi ad uno spazio di indirizzamento virtuale, indipendente dall'effettivo spazio di indirizzamento della memoria fisica.
- Si genera una macchina virtuale in cui ogni processore (virtuale) ha a disposizione una memoria privata la cui corrispondenza con la memoria fisica non è (in linea di principio) rilevante.
- I programmi possono essere sviluppati senza sapere la configurazione reale della memoria in cui saranno allocati

## Gestore della memoria centrale (2)

- Alla base di questo livello di virtualizzazione ci sono i meccanismi di
  - rilocalizzazione

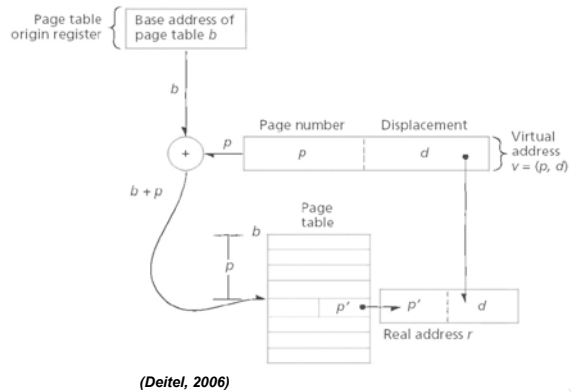


(Stallings, 2011)

## Gestore della memoria centrale (3)

- Alla base di questo livello di virtualizzazione ci sono i meccanismi di

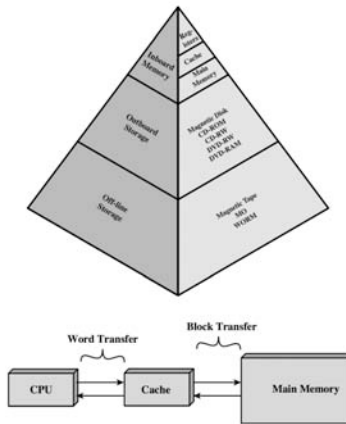
- rilocalizzazione
- paginazione



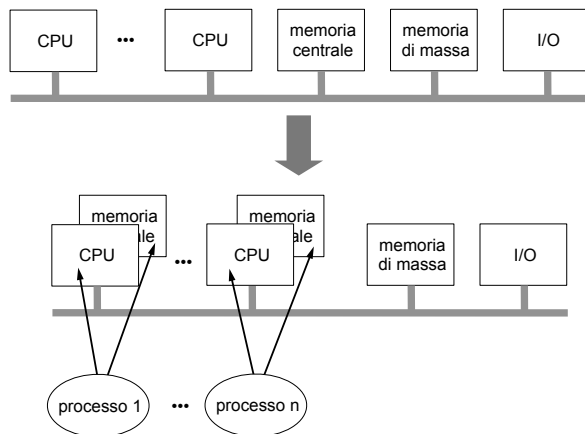
## Gestore della memoria centrale (4)

- Alla base di questo livello di virtualizzazione ci sono i meccanismi di

- rilocalizzazione
- paginazione
- gerarchie di memoria



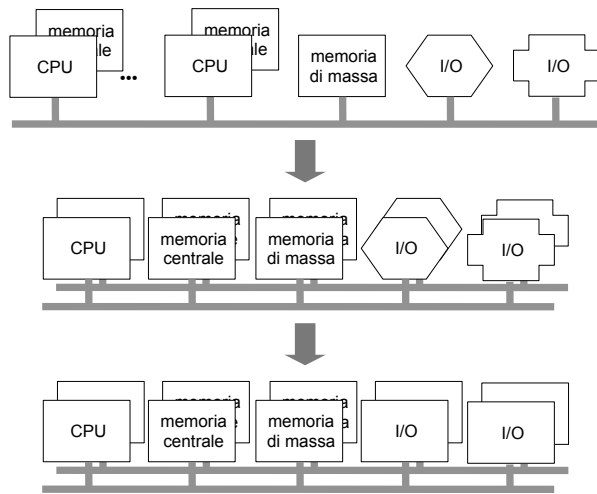
## Gestore della memoria centrale (5)



## Gestore delle periferiche di I/O (1)

- Il gestore delle periferiche realizza due tipi di virtualizzazione:
  - come i due gestori precedenti, ripartisce l'utilizzo delle risorse esterne in modo che ogni processo possa vedere una periferia dedicata, in cui
    - non ci sono conflitti di utilizzo con altri processi o utenti
    - non è necessario gestire la sincronizzazione e i tempi di attesa (operazioni sincrone)
  - fornisce un insieme di *driver* con funzioni di gestione di alto livello che, mascherando le differenze costruttive delle apparecchiature, rendono omogeneo l'utilizzo della periferia anche se questa è diversificata
- Alla base di questo livello di virtualizzazione ci sono i meccanismi di I/O a interrupt e in DMA

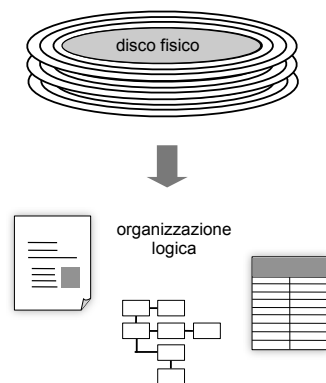
## Gestore delle periferiche di I/O (2)



## File system (1)

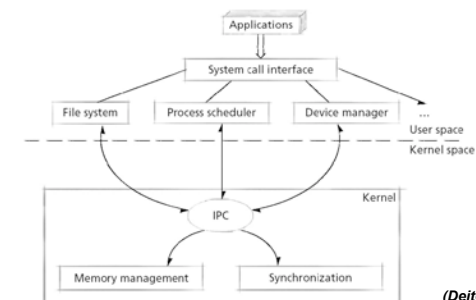
- Il file system assegna alle informazioni memorizzate su una memoria di massa una organizzazione e una strutturazione riferite all'utilizzo delle informazioni e non al loro schema di memorizzazione
  - le informazioni sono raccolte in unità logiche (*file*) identificate da un nome, di cui si ignora la struttura fisica e l'allocazione sul supporto di memorizzazione
  - l'accesso avviene tramite funzioni che operano sul contenuto in base alla sua organizzazione logica (*record, stream*)
  - sono gestite sia la privacy dei dati, sia i conflitti di accesso nel caso di utilizzo condiviso tra più utenti

## File system (2)



## Sistemi operativi a microkernel

- Dividono i servizi in più processi che comunicano attraverso un nucleo semplificato
  - il nucleo implementa solo le funzioni più critiche, che devono essere eseguite in kernel mode
  - la struttura risultante è altamente modificabile e adattabile



(Deitel, 2005)

## Aumento della complessità dei sistemi operativi (1)

dimensione del nucleo di alcuni sistemi operativi storici

Year	AT&T	BSD	MINIX	Linux	Solaris	Win NT
1976	V6 9K					
1979	V7 21K					
1980		4.1 38K				
1982	Sys III 58K					
1984		4.2 98K				
1986		4.3 179K				
1987	SVR3 92K		1.0 13K			
1989	SVR4 280K					
1991				0.01 10K		
1993		Free 1.0 235K			5.3 850K	3.1 6M
1994		4.4 Lite 743K		1.0 165K		3.5 10M
1996				2.0 470K		4.0 16M
1997			2.0 62K		5.6 1.4M	
1999				2.2 1M		
2000		Free 4.0 1.4M			5.8 2.0M	2000 29M

(Tanenbaum, 2001)



## Aumento della complessità dei sistemi operativi (2)

Operating System	Release Date	Minimum Memory Requirement	Recommended Memory
Windows 1.0	November 1985	256KB	
Windows 2.03	November 1987	320KB	
Windows 3.0	March 1990	896KB	1MB
Windows 3.1	April 1992	2.6MB	4MB
Windows 95	August 1995	8MB	16MB
Windows NT 4.0	August 1996	32MB	96MB
Windows 98	June 1998	24MB	64MB
Windows ME	September 2000	32MB	128MB
Windows 2000 Professional	February 2000	64MB	128MB
Windows XP Home	October 2001	64MB	128MB
Windows XP Professional	October 2001	128MB	256MB (Deitel, 2005)

