

Sistemi Operativi

Rocco Aversa

Tel. 0815010268

rocco.aversa@unina2.it

Ricevimento: Ma 14.00-16.00

Gi 14.00-16.00

1

Libri di Testo

- **Testo Principale**

- A.S. Tanenbaum, *I moderni sistemi operativi (3 ed)*, Prentice-Hall 2009
- A.S. Tanenbaum, *Modern Operating Systems (2 ed)*, Prentice-Hall 2002

- **Approfondimenti**

**Lucidi dalle lezioni e materiale vario
disponibili di volta in volta sul sito del corso:**

<http://research.diii.unina2.it/raversa/sistemioperativi/>

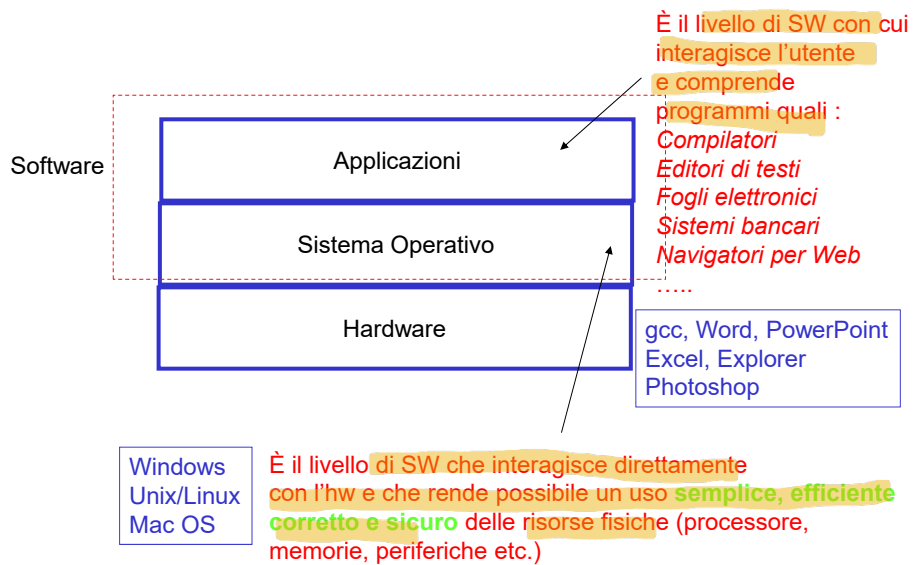
2

Introduzione

Cos'è un sistema operativo ?

3

Cos'è un sistema operativo ?



4

Semplice: offrire a utente finale una macchina che apparentemente sa fare cose più complesse di quello che il processore sa fare (senza so devo usare Assembly)

Cosa è un sistema operativo?

- **Macchina Virtuale di livello 3:** estende il livello sottostante fornendo operazioni astratte per l'interazione con i dispositivi di I/O e per l'esecuzione contemporanea di più programmi.
- **Gestore di risorse:** coordina l'uso condiviso delle risorse del sistema (CPU, memorie, dischi, stampanti, ecc.) da parte di più programmi (in esecuzione simultanea) cercando di utilizzarle al meglio (minimizzare i tempi morti).
- **Controllore:** controlla che un programma in esecuzione non acceda ad informazioni di altri programmi in esecuzione o del sistema operativo, compromettendone la riservatezza o il buon funzionamento

Ottimizzazioni hanno un costo: es. deve tenere conto di dove si è fermato con certo processo. Ma servono spazi per strutture dati che gestiscono la multiprogrammazione

5

Macchina virtuale di livello 3



6

Quali sono le funzioni di un SO ? (Estensioni)

- Esegue applicazioni :
 - carica il programma binario prodotto della compilazione (e residente su disco) nella RAM,
 - cede il processore all'applicazione da eseguire
- Facilita l'accesso alle periferiche/dispositivi
 - mette a disposizione operazioni di lettura/scrittura, invio/ricezione dati ad alto livello che possono essere usate senza conoscere i dettagli tecnici della periferica

7

Quali sono le funzioni di un SO ? (2) (Estensioni)

- Archivia dati e programmi :
 - mette a disposizione dell'utente una visione astratta della memoria secondaria (il file system basato sulle astrazioni : *file/archivi* e *folder/cartelle*)
 - gestisce la realizzazione di queste astrazioni sul supporto fisico (disco) gestendo tutti i dettagli legati alla lettura/scrittura dei settori



8

Quali sono le funzioni di un SO ? (3)

(Gestione)

- **Gestisce le risorse**
 - Ripartisce, secondo opportune politiche, le risorse disponibili (processore, RAM, periferiche) fra le varie applicazioni/utenti
 - evita che ci siano malfunzionamenti dovuti all'uso contemporaneo di risorse
 - es: un word processor e un web browser che inviano contemporaneamente dati alla stampante provocano una stampa erronea
 - ottimizza le prestazioni scegliendo delle politiche che permettano di sfruttare al meglio tutte le parti del computer



9

Quali sono le funzioni di un SO ? (4)

(Gestione e controllo)

- **Gestisce malfunzionamenti del sistema**
 - **rileva e gestisce situazioni anomale**
 - es: (1) se il disco ha un settore difettoso, il SO può ricopiare le informazioni residenti su quel settore da un'altra parte (in modo trasparente all'utente)
- **Controlla che non ci siano interazioni potenzialmente dannose tra diversi programmi e utenti**
 - es: se un'applicazione cerca di effettuare una operazione non permessa (come leggere i dati di un'altra applicazione) il SO può bloccare l'applicazione segnalando all'utente la situazione

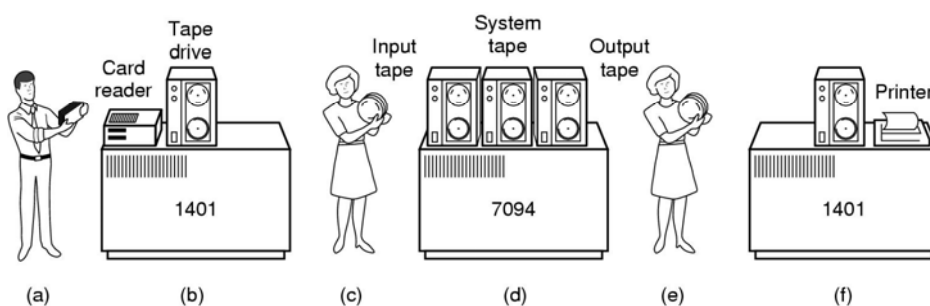
10

Un po' di storia

- **Prima generazione 1945 - 1955**
 - valvole, programmi “hardwired”
- **Seconda generazione 1955 - 1965**
 - transistor, sistemi batch
- **Terza generazione 1965 – 1980**
 - Circuiti integrati (IC) e multiprogramming
- **Quarta generazione 1980 – oggi**
 - personal computers, reti

11

La seconda generazione: sistemi batch



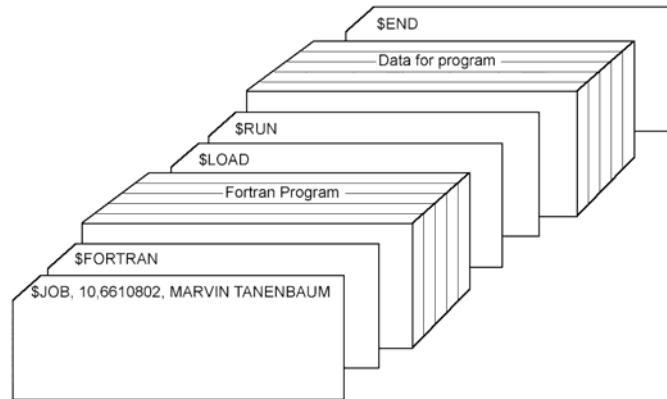
Nei primi sistemi batch l'operatore:

- porta le schede al 1401 che trasferisce il contenuto delle schede su nastro
- inserisce il nastro nel 7094 (che effettua la vera e propria computazione)
- inserisce il nastro su un altro 1401 che stampa i risultati

Prime idee: disaccoppiare I/O lento dalla computazione

12

Struttura di un job (seconda-terza generazione)



Sistemi **NON INTERATTIVI**: un operatore si occupa di far eseguire i job

L'unica componente di sistema operativo erano le routine per gestire l'I/O

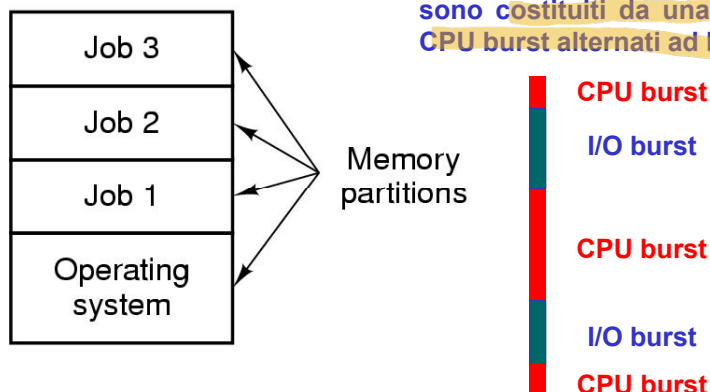
13

La terza generazione e la multiprogrammazione

L'avvento dei dischi ha permesso di escludere i nastri: tecnica dello **SPOOLING** (Simultaneous Peripheral Operation On-Line).

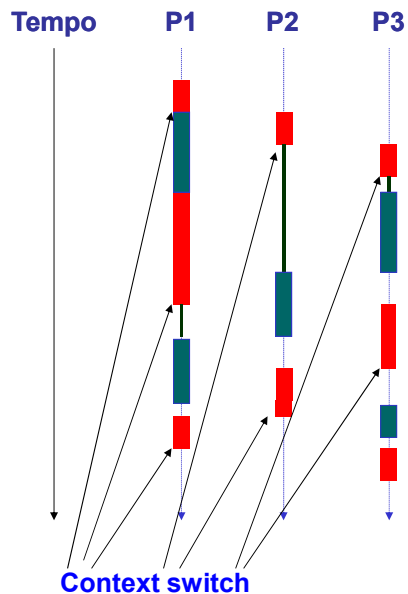
Obiettivo: utilizzare al massimo la CPU evitando i tempi morti dovuti alla lentezza dell'I/O rispetto alla CPU

I job (programmi) in esecuzione sono costituiti da una sequenza di CPU burst alternati ad I/O burst



14

Esecuzione in un sistema multiprogrammato



La CPU viene assegnata a turno ai vari programmi in esecuzione: quando il programma a cui è assegnata la CPU chiede (al S.O.) l'esecuzione di una operazione di I/O, la CPU viene assegnata ad un altro programma. Se il livello di multiprogrammazione è alto, la CPU non sarà mai inutilizzata.

- Indica chi è *running* in ogni istante
- Indica per chi lavora il disco in ogni istante
- Indica un processo in attesa di iniziare I/O

15

La multiprogrammazione: vantaggi e problemi

Vantaggi: mentre un programma deve attendere la fine di una operazione di I/O, un altro può essere eseguito, quindi la CPU viene sfruttata meglio.

Problemi: Bisogna far convivere diversi programmi, appartenenti anche ad utenti diversi, senza che interferiscano. Occorre gestire in modo appropriato le risorse (CPU e Memoria, in primo luogo) e assicurare una esecuzione protetta.

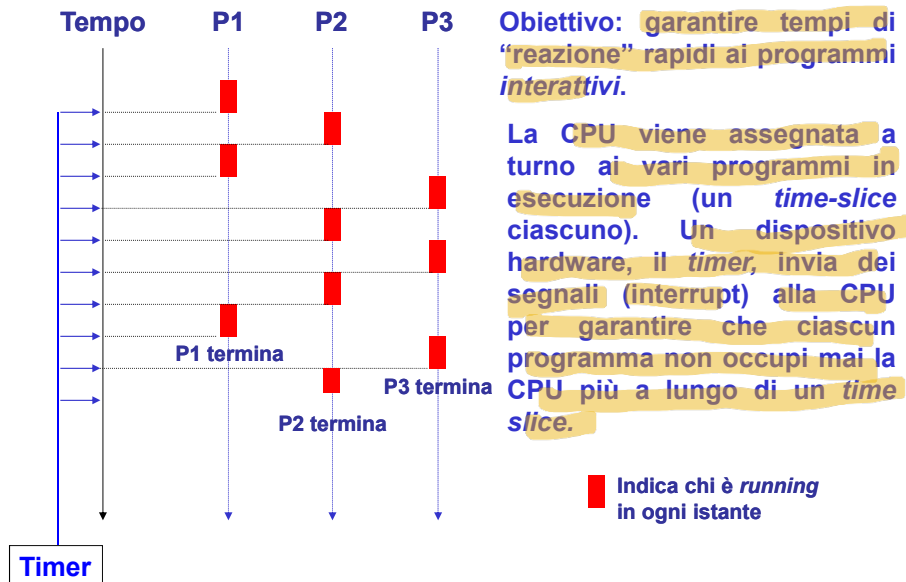
Esempio (context switch):

trasferimento della CPU da un programma ad un altro; ad ogni evento di questo tipo, il S.O. deve salvare lo stato del programma (i registri della CPU) in esecuzione e caricare lo stato salvato del programma prescelto tra quelli in attesa della CPU.

↳ es. Registro di stato con *iflag*

16

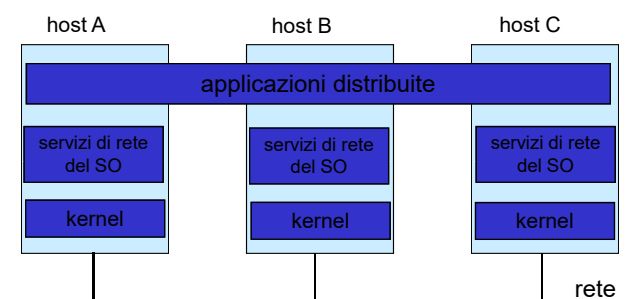
Dalla multiprogrammazione al time sharing



17

Sistemi Operativi in rete

- Oltre ai compiti tradizionali di un Sistema Operativo vengono fornite al programmatore ulteriori funzionalità per consentire l'utilizzo di risorse disponibili in rete.



Come comunicano processi differenti? Pacchetti

18

Tipologie di Sistemi Operativi (1)

- **S.O. per mainframe (es: IBM sistema z/OS)**

Calcolatori con capacità di I/O molto elevata (molti dischi veloci e enorme spazio – centinaia di Gbyte e oltre). Vengono utilizzati come server per grandi aziende o come Web server capaci di servire tantissime richieste simultanee. Gestiscono un misto di job batch, transaction processing, time sharing.

- **S.O. per Server (es: UNIX, Windows NT/Windows Server 2003/2008)**

Sono fatti per servire molte richieste provenienti da diversi utenti collegati in rete. Rendono possibile la condivisione di risorse di vario tipo (print server, file server, mail server, file server).

- **S.O. per PC/Workstation: Windows 95/98/2000/XP/Vista/7/8/10 Linux, MacOS**

A differenza dei precedenti, orientati a servire molti utenti, questi sono orientati a servire un singolo utente (o almeno un utente alla volta).

19



Tipologie di Sistemi Operativi (2)

- **S.O. per sistemi real time (hard e soft): QNX, VxWorks, UnixRT(soft)**

catene di montaggio ecc.
Sono caratterizzati dalla necessità di soddisfare determinate **deadline**: *hard* se mancare la deadline può fare danni (controllo di un robot in una catena di montaggio), *soft* se mancare la deadline significa solo fornire un servizio degradato (es. streaming audio/video) *Hard: scadenze inderogabili*

- **S.O. per sistemi embedded. Windows C.E., Android, IOS, PalmOS, legOS**

- Smartphone
- PDA (Personal Digital Assistant)
- sistemi "on board"
- Lego Mindstorms.



- **S.O. per smart card (rudimentali: pochissimo spazio)**

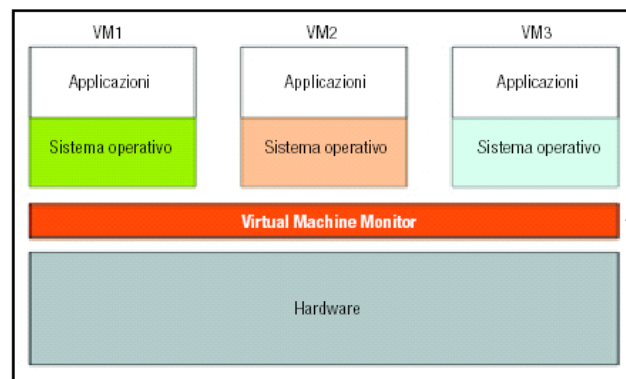
- pagamenti elettronici
- Java-oriented: eseguono una JVM, possono ricevere applet ed eseguirli

20

Virtualizzazione

(cloud computing ha portato una spinta: provider che offre risorse simili le capacità elaborative dove

- Sullo stesso elaboratore possono essere eseguiti indipendentemente e anche parallelamente sistemi operativi diversi (es. VMware, Microsoft Hyper-V, Virtual PC, Xen, Virtual Box)
- Il VMM replica per ogni macchina virtuale le stesse interfacce hardware funzionalmente identiche a quelle della sottostante macchina fisica



→ livello aggiunto che divide le risorse hardware e le fornisce al sistema operativo

21

Questo porta inevitabilmente a degradazione delle qualità e efficienza.