



Università degli Studi di Palermo  
*Dipartimento di Ingegneria Informatica*



# Elaborazione di Immagini e Suoni / Riconoscimento e Visioni Artificiali 12 c.f.u.

Anno Accademico 2008/2009

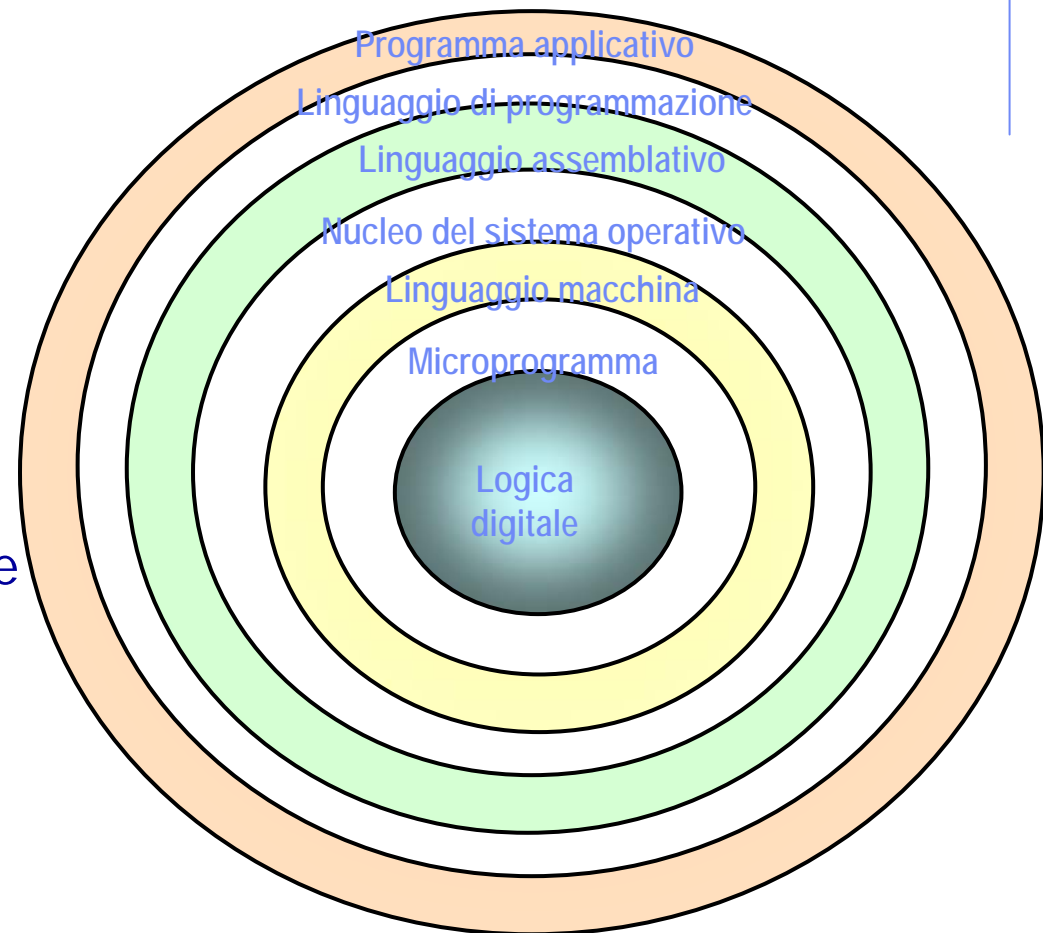
Docente: ing. Salvatore Sorce

## Il Sistema Operativo

Facoltà di Lettere e Filosofia

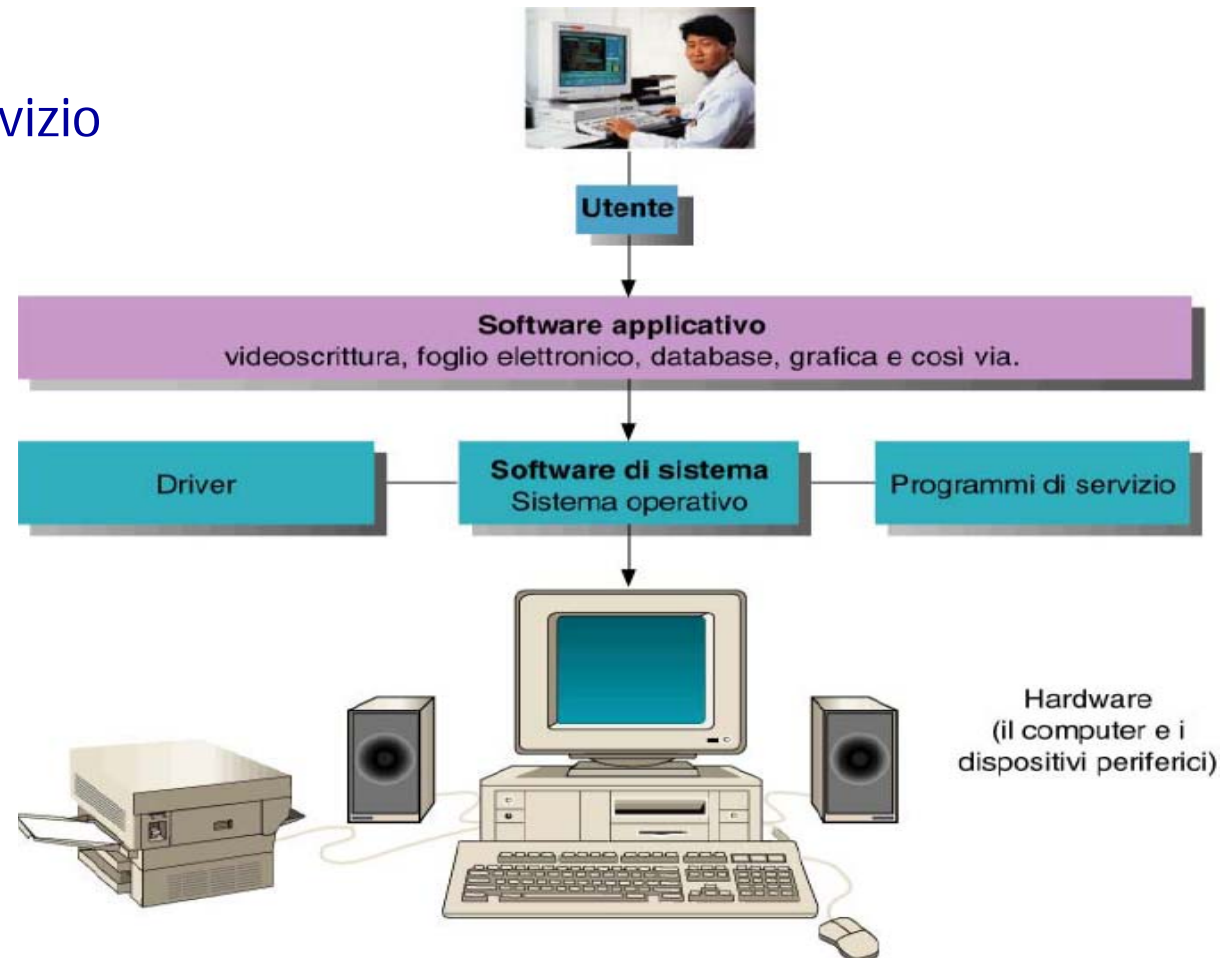
## Gerarchia del software

- Sei livelli di astrazione separano l'utente dall'hardware sottostante
- Microprogramma
- Linguaggio macchina
- Sistema operativo
- Linguaggio assembler
- Linguaggio di programmazione
- Programma applicativo



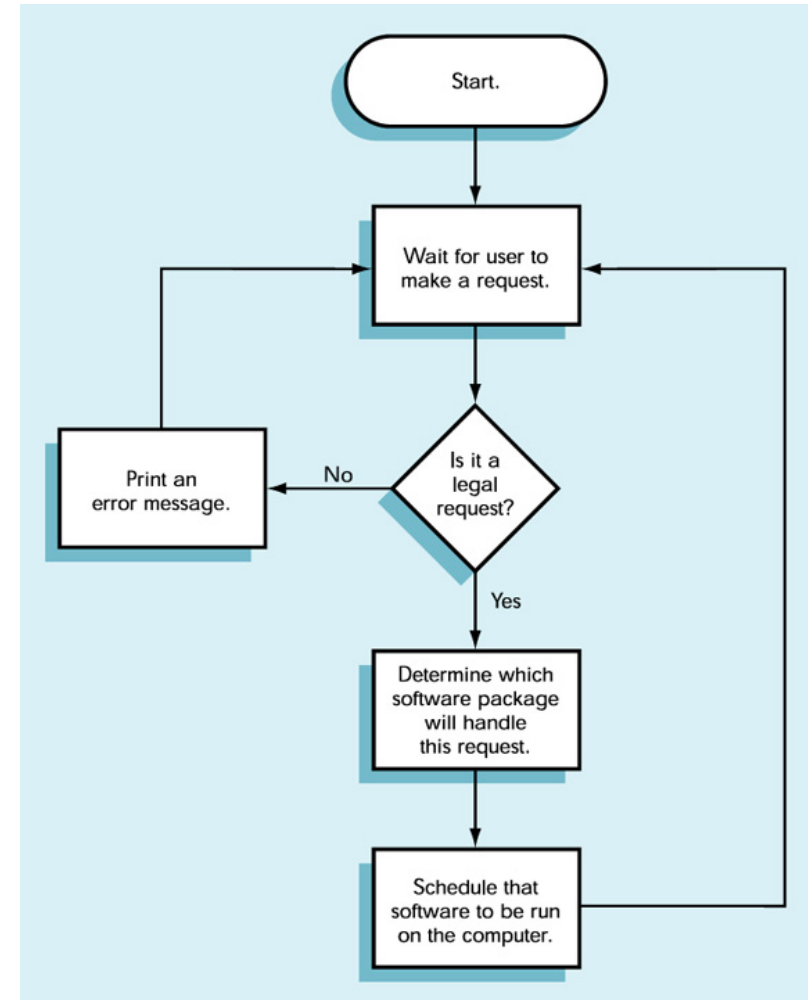
## Software di sistema

- Sistemi operativi
  - Windows, DOS, Unix/Linux, Mac OS
- Driver
- Programmi di servizio



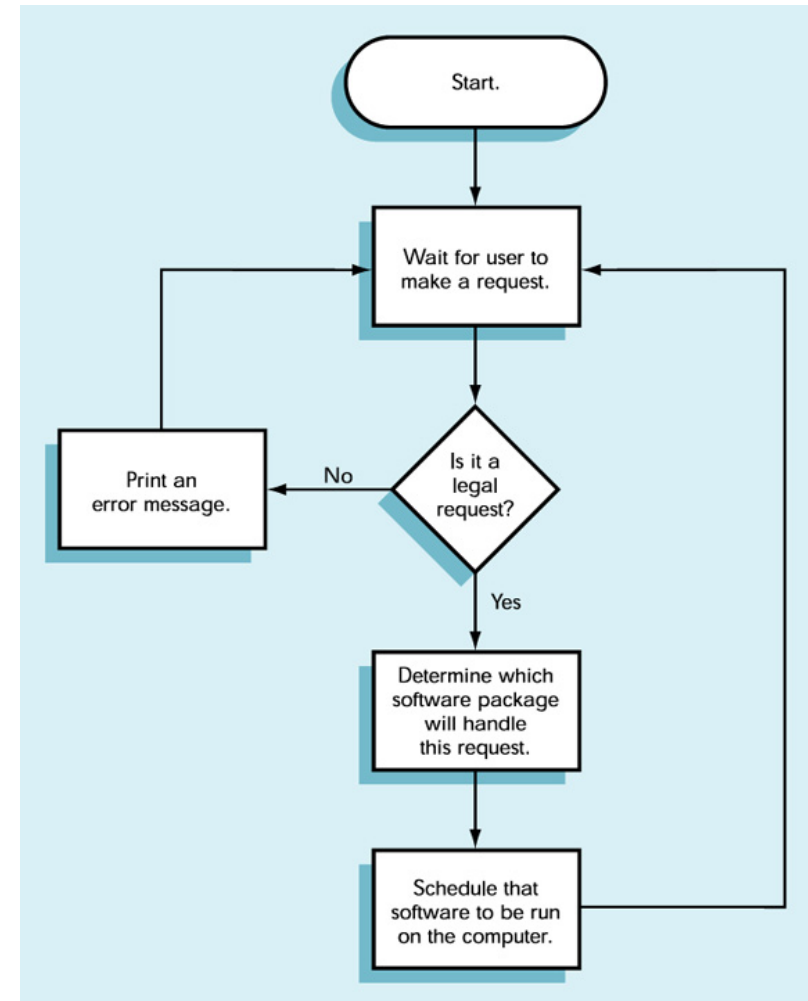
## Sistema operativo

- Per realizzare i servizi di editing, assemblaggio ed esecuzione di un programma occorre eseguire comandi di sistema
  - >assemble myProg
  - >run myProg
  
- Funzioni del sistema operativo
  - Interfaccia utente
  - Scelta e attivazione del programma da eseguire
  - Protezione e sicurezza del sistema
  - Allocazione efficiente delle risorse
  - Utilizzo sicuro delle risorse



## Sistema operativo

- Viene invocato ogniqualvolta il processore non è in uso da altri componenti sw
- Gestisce l'interazione utente-calcolatore
- Tipici comandi
  - Tradurre un programma
  - Caricare un programma in memoria
  - Collegare parti di sw per costruire un unico programma
  - Eseguire un programma
  - Salvare informazioni
  - Elencare file
  - Stampare file
  - Stabilire connessione di rete
  - Mantenere e comunicare data e ora correnti





## Interfacce utenti

- Implementano i comandi del SO
- Interfacce testuali
  - Uso di un linguaggio di comandi immessi come testo da tastiera

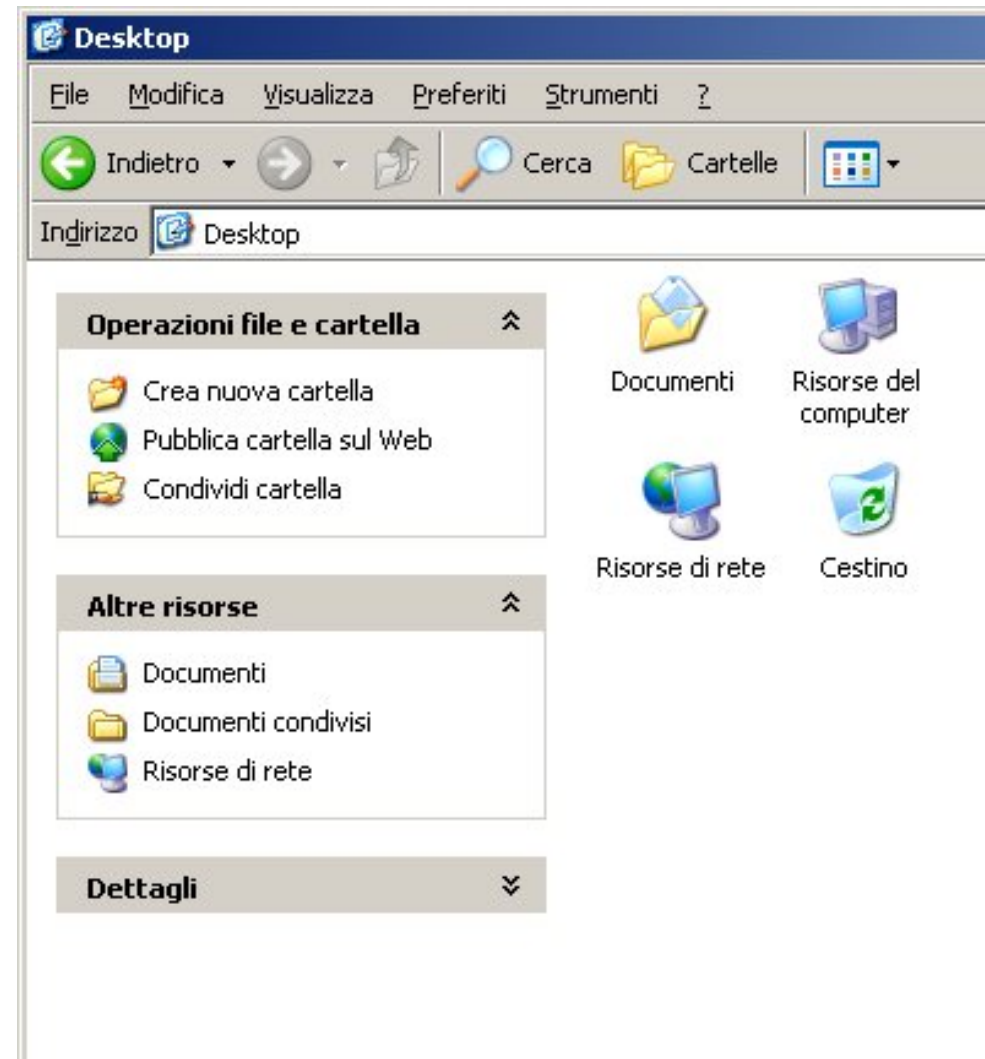
```
* * * * *  
*  
* Welcome to Apollo ...  
*  
* Macalester College AlphaServer 2000 *  
*  
* * * * *  
  
User Name:  Schneider  
Password:  XXXXX   (Blocked out for security reasons)  
$          ($ is the prompt. The system is now waiting for a user request.)
```

```
Microsoft Windows XP [Versione 5.1.2600]  
D:\>
```

## Interfacce utenti

### ➤ Interfacce grafiche (GUI)

- Uso di una metafora (desktop)
- Ai comandi testuali sono sostituiti icone, menu, finestre e le azioni che possono essere eseguite su di essi



## File system

- Esistono diversi tipi di supporti per la memorizzazione permanente delle informazioni: dischi magnetici (floppy disk, hard disk), dischi ottici (cd), nastri magnetici
- Un *file* è un insieme di byte che rappresentano una certa entità logica (testo, immagine, suono, programma, etc), organizzati secondo un certo formato, memorizzati su supporti di memoria secondaria.



## File system

- Il **File System** è quella parte del S.O. che si occupa di gestire e strutturare le informazioni memorizzate su supporti permanenti
  - Il sistema operativo deve fornire una visione **astratta** (semplificata) dei file su disco e l'utente deve avere la possibilità di:
    - identificare ogni file con un nome (**filename**) astraendo completamente dalla sua memorizzazione fisica (blocchi su disco rigido e localizzazione dei blocchi)
    - avere un insieme di operazioni per lavorare sui file: creare o rimuovere un file, copiarlo, cambiargli nome, inserire informazioni in un file
    - effettuare l'accesso alle informazioni mediante operazioni ad alto livello, che non tengono conto del tipo di memorizzazione (accedere ad un file memorizzato sul disco rigido oppure su un CD-ROM allo stesso modo)
- (segue ...)

## File system

(... segue)

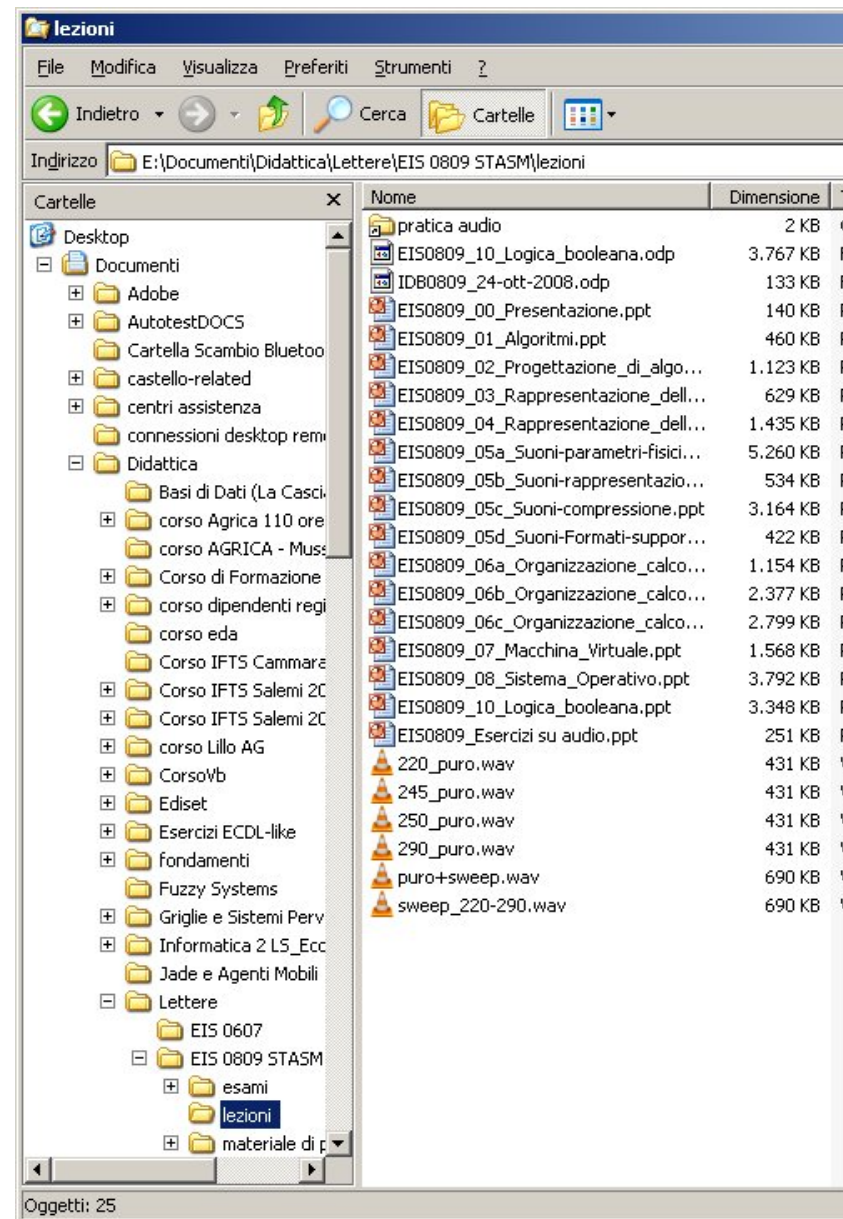
- avere la possibilità di strutturare un insieme di file, organizzandoli in sottoinsiemi secondo le loro caratteristiche, per avere una visione ordinata e strutturata delle informazioni sul disco
  - in un sistema multi-utente, inoltre l'utente deve avere meccanismi per proteggere i propri file, ossia per impedire ad altri di leggerli, scriverli o cancellarli
- i moderni sistemi operativi forniscono supporto per queste attività

## File system

- Il file system deve tenere traccia di tutte le caratteristiche di file e sottoinsiemi di file (il nome, la dimensione, quali sono gli indirizzi dei blocchi sui quali sono memorizzati, etc.)
- Dove sono memorizzate queste informazioni?
- Una parte del disco rigido (un sottoinsieme di tracce) è riservato al sistema operativo per questi (ed altri) scopi
- Esempio: FAT (File Allocation Table)
  - Contiene le corrispondenze <nome file> → <blocco di inizio>
  - Settori concatenati: <1° blocco file> → <2° blocco file> → ...
- N.B. Anche una parte della memoria centrale (RAM) è riservata alla memorizzazione del sistema operativo

## File system

- Presentazione dei file all'utente
  - Directory (cartelle)
  - Organizzazione gerarchica ad albero
  - Nomi dei files
  - Percorsi





## Allocazione efficiente delle risorse

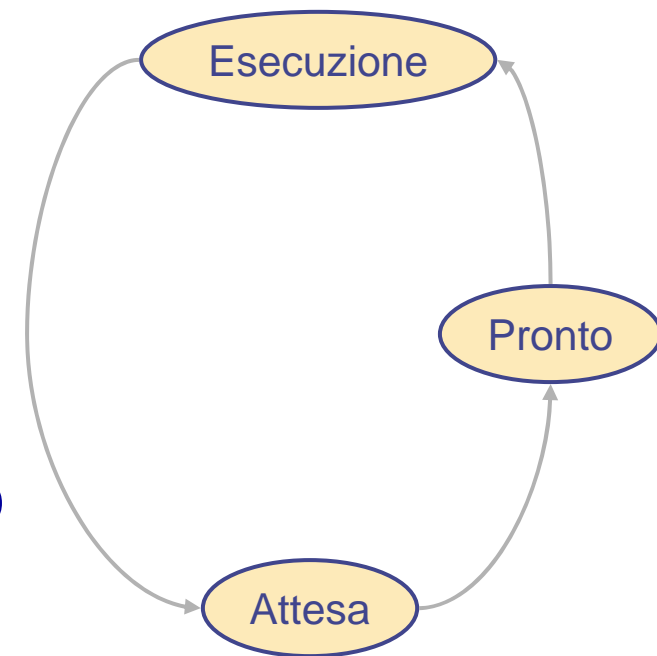
- Esiste una notevole differenza nella velocità di esecuzione di operazioni tra processore e unità di I/O
- Il SO deve assicurare che il processore rimanga inutilizzato il minor tempo possibile
- Tanti programmi in esecuzione, ma un solo processore: **quasi parallelismo**
- Il SO mantiene una coda di programmi in esecuzione dei quali solo uno è attivo per ogni istante di tempo



## Allocazione efficiente delle risorse

### ➤ stati di un programma

- In esecuzione  
programma attualmente in esecuzione
- Pronto  
programmi in memoria e pronti per l'esecuzione, ordinati per priorità
- Attesa  
programmi che non possono essere eseguiti perché in attesa del completamento di una operazione di I/O





## Allocazione efficiente delle risorse

- Quattro programmi, A, B, C, D
  - A in esecuzione
  - B, C, D pronti per passare in esecuzione

Attesa	Pronto	Esecuzione
	B	A
	C	
	D	



## Allocazione efficiente delle risorse

- Quattro programmi, A, B, C, D
  - A in esecuzione
  - B, C, D pronti per passare in esecuzione
- A inizia una operazione di I/O
  - A passa in attesa e B va in esecuzione

Attesa	Pronto	Esecuzione
A	C	B
	D	





## Allocazione efficiente delle risorse

- Quattro programmi, A, B, C, D
  - A in esecuzione
  - B, C, D pronti per passare in esecuzione
- A inizia una operazione di I/O
  - A passa in attesa e B va in esecuzione
- B inizia una operazione di I/O
  - B passa in attesa e C va in esecuzione

Attesa	Pronto	Esecuzione
A	D	C
B		



## Allocazione efficiente delle risorse

- Quattro programmi, A, B, C, D
  - A in esecuzione
  - B, C, D pronti per passare in esecuzione
- A inizia una operazione di I/O
  - A passa in attesa e B va in esecuzione
- B inizia una operazione di I/O
  - B passa in attesa e C va in esecuzione
- A completa l'operazione
  - Passa in pronto. Se ha priorità superiore a D, potrebbe scavalcarlo

Attesa	Pronto	Esecuzione
B	D	C
	A	



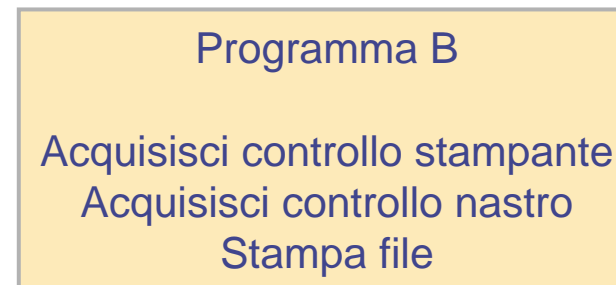
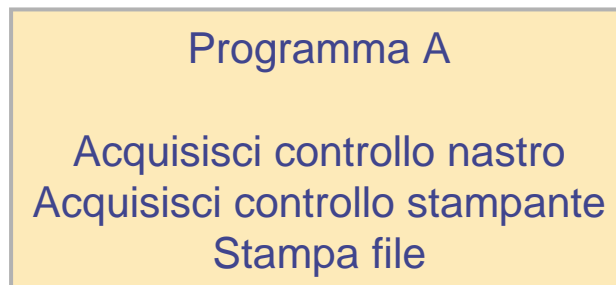
## Allocazione efficiente delle risorse

- Quattro programmi, A, B, C, D
  - A in esecuzione
  - B, C, D pronti per passare in esecuzione
- A inizia una operazione di I/O
  - A passa in attesa e B va in esecuzione
- B inizia una operazione di I/O
  - B passa in attesa e C va in esecuzione
- A completa l'operazione
  - Passa in pronto. **Se ha priorità superiore a D, potrebbe scavalcarlo**

Attesa	Pronto	Esecuzione
B	A	C
	D	

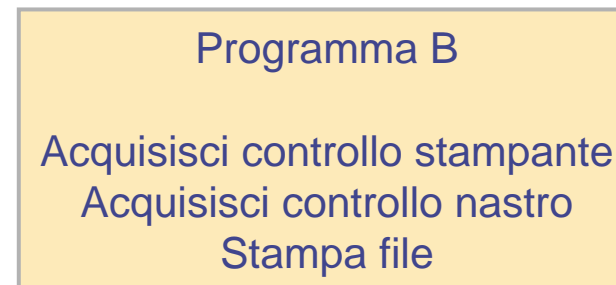
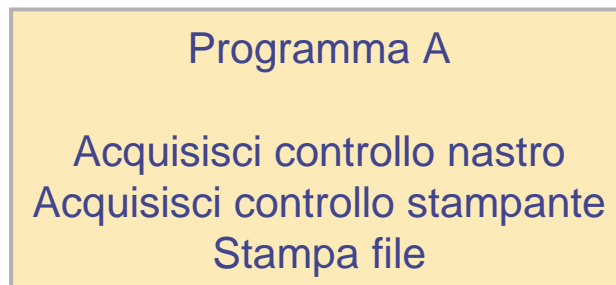
## Utilizzo sicuro delle risorse

- Le risorse devono essere utilizzate in sicurezza, ovvero evitando che possano portare il sistema in **stallo** (*deadlock*)
- Esempio: due programmi in esecuzione, A e B, devono caricare un dato da nastro e stamparlo su una stampante laser
- L'ordine di acquisizione delle risorse può determinare un deadlock



## Utilizzo sicuro delle risorse

- Le risorse devono essere utilizzate in sicurezza, ovvero evitando che possano portare il sistema in **stallo** (*deadlock*)
- Esempio: due programmi in esecuzione, A e B, devono caricare un dato da nastro e stamparlo su una stampante laser
- L'ordine di acquisizione delle risorse può determinare un deadlock
  - A va in deadlock perché possiede controllo solo sulla prima risorsa, mentre l'altra risorsa è bloccata da B, anch'esso in deadlock





## Utilizzo sicuro delle risorse

- Le risorse devono essere utilizzate in sicurezza, ovvero evitando che possano portare il sistema in **stallo** (*deadlock*)
- Esempio: due programmi in esecuzione, A e B, devono caricare un dato da nastro e stamparlo su una stampante laser
- L'ordine di acquisizione delle risorse può determinare un deadlock
  - A va in deadlock perché possiede controllo solo sulla prima risorsa, mentre l'altra risorsa è bloccata da B, anch'esso in deadlock
  - I due programmi passano in modo di attesa permanente, perché l'uno aspetta una risorsa bloccata dall'altro

## Panoramica storica

Generazione	Date	Caratteristiche
Prima	1945-1955	Nessun Sistema operativo Programmi direttamente costruiti sulle macchine ed eseguiti manualmente
Seconda	1955-1965	Sistemi operativi batch Concetto di utilizzazione del sistema Linguaggio di controllo dei job
Terza	1965-1985	Sistemi operativi multiprogramma, sistemi operativi time-sharing Concetto di protezione dei dati e sicurezza di esecuzione Utilizzo interattivo del calcolatore Autenticazione utenti e controllo di accesso Primo sistema operativo per PC
Quarta	1985-Oggi	Sistemi operativi in rete, LAN File server, print server, server di elaborazione Accesso remoto, e-mail Interfacce utente grafiche Sistemi operativi in tempo reale, sistemi operativi embedded
Quinta	??	Interfacce utente multimediali Sistemi operativi paralleli Ambienti di elaborazione distribuiti