



Università degli Studi di Palermo
Dipartimento di Ingegneria Informatica



Informatica di Base - 6 c.f.u.

Anno Accademico 2007/2008

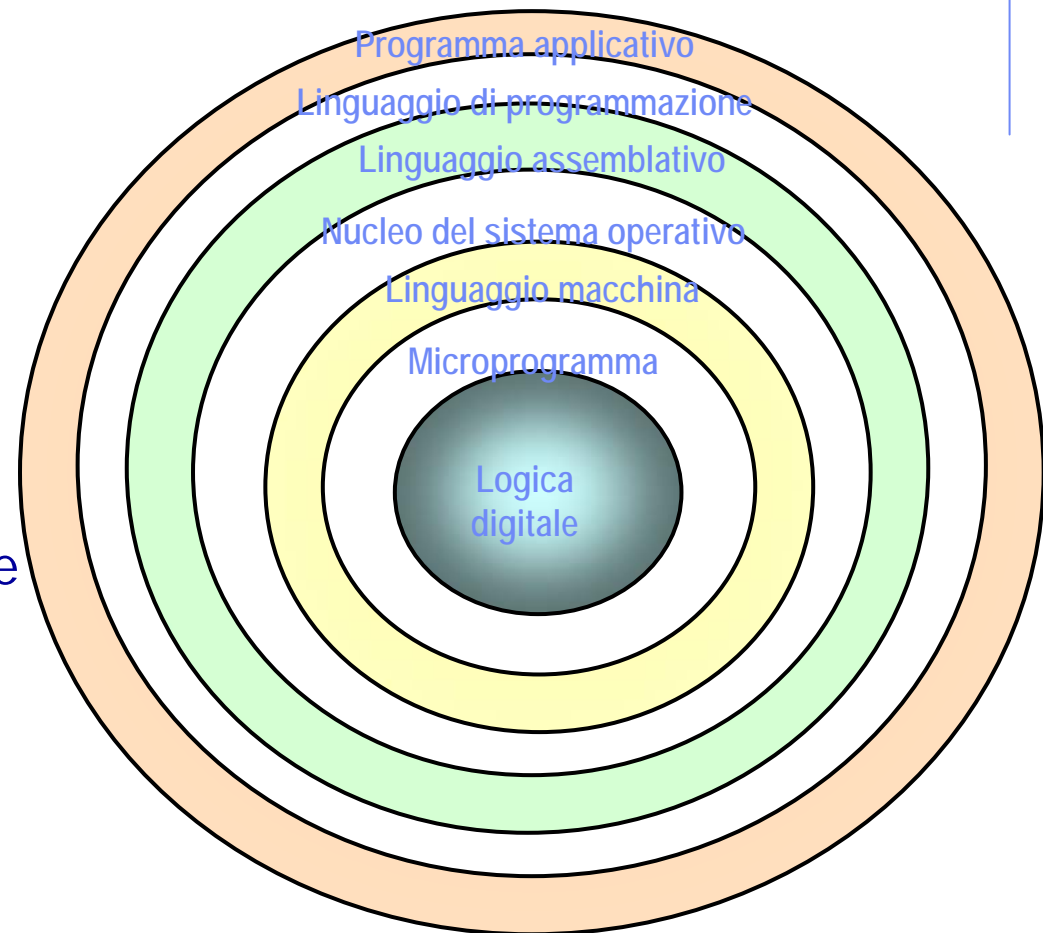
Docente: ing. Salvatore Sorce

Il Sistema Operativo

Facoltà di Lettere e Filosofia

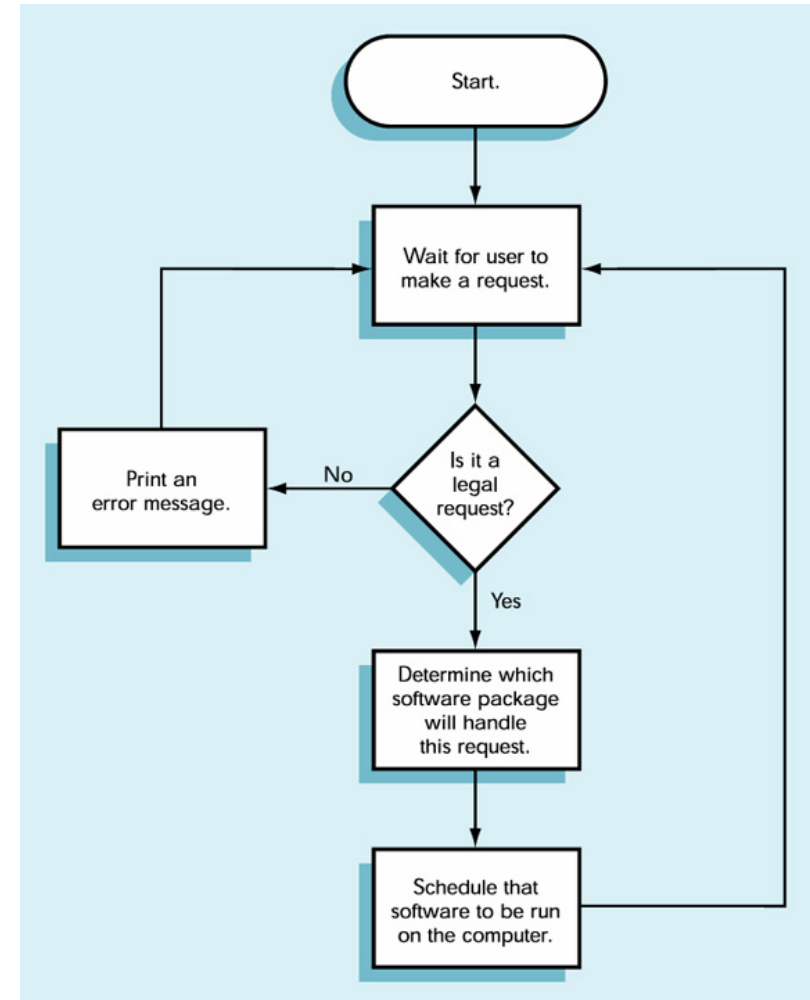
Gerarchia del software

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



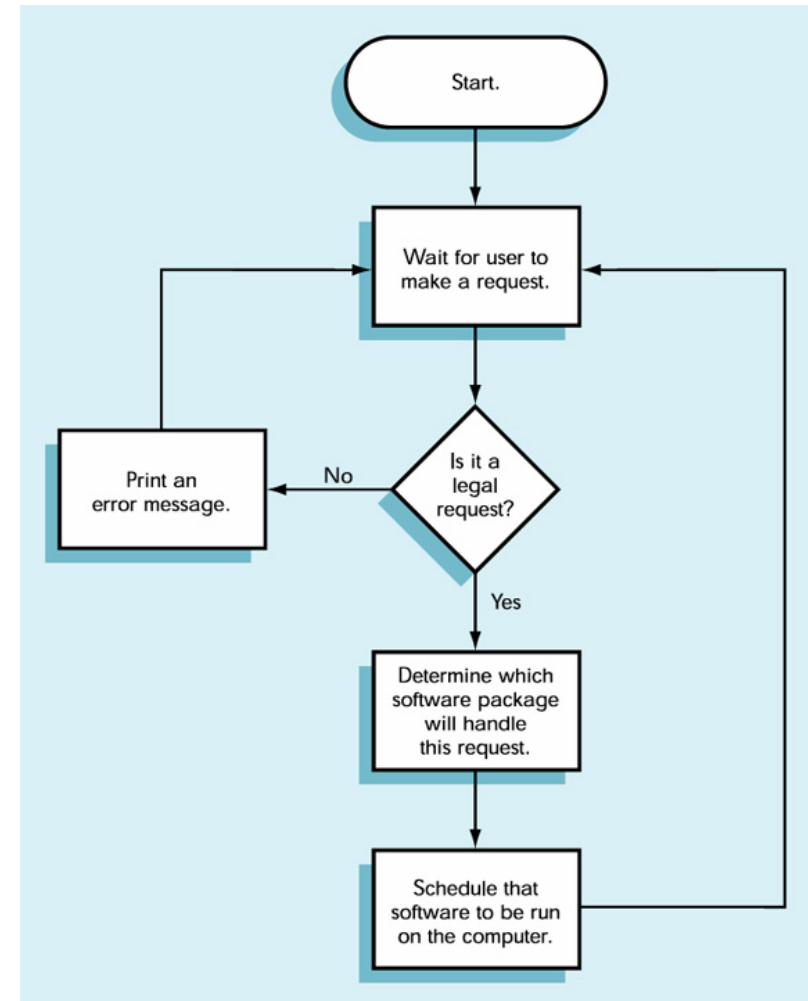
Sistema operativo

- 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

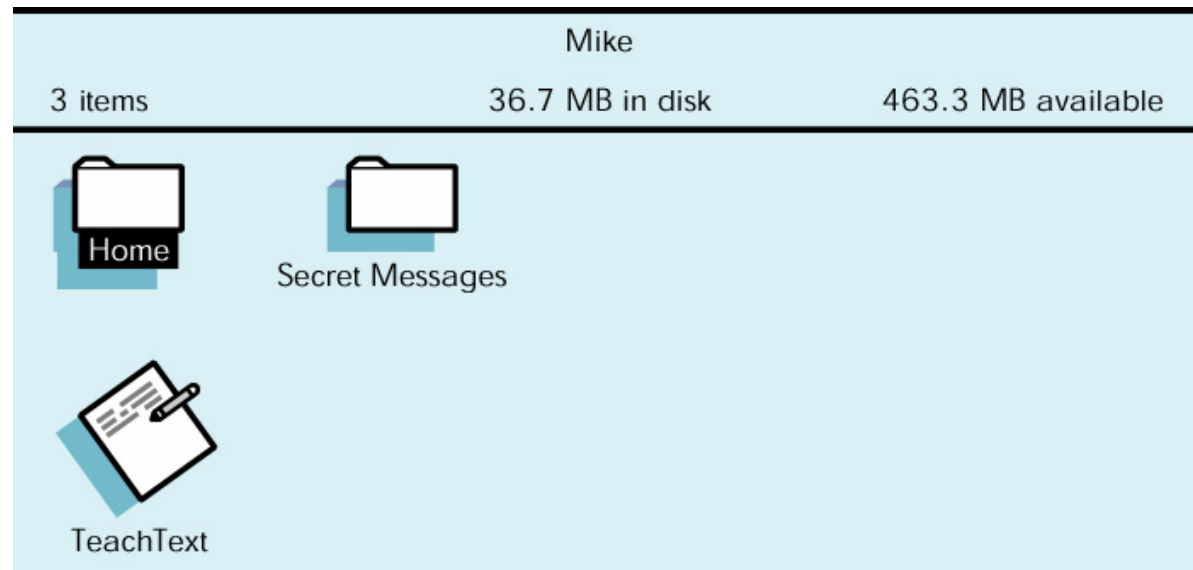
- Interfacce utente
 - Implementano i comandi del SO
- Interfacce testuali
 - Uso di un linguaggio di comandi
- Interfacce grafiche (GUI)
 - Uso di una metafora (desktop)
 - Ai comandi sono sostituiti icone, menu, finestre e le azioni che possono essere eseguite su di essi

```

* * * * *
*
* Welcome to Apollo ...
*
* Macalester College AlphaServer 2000
*
* * * * *
    
```

```

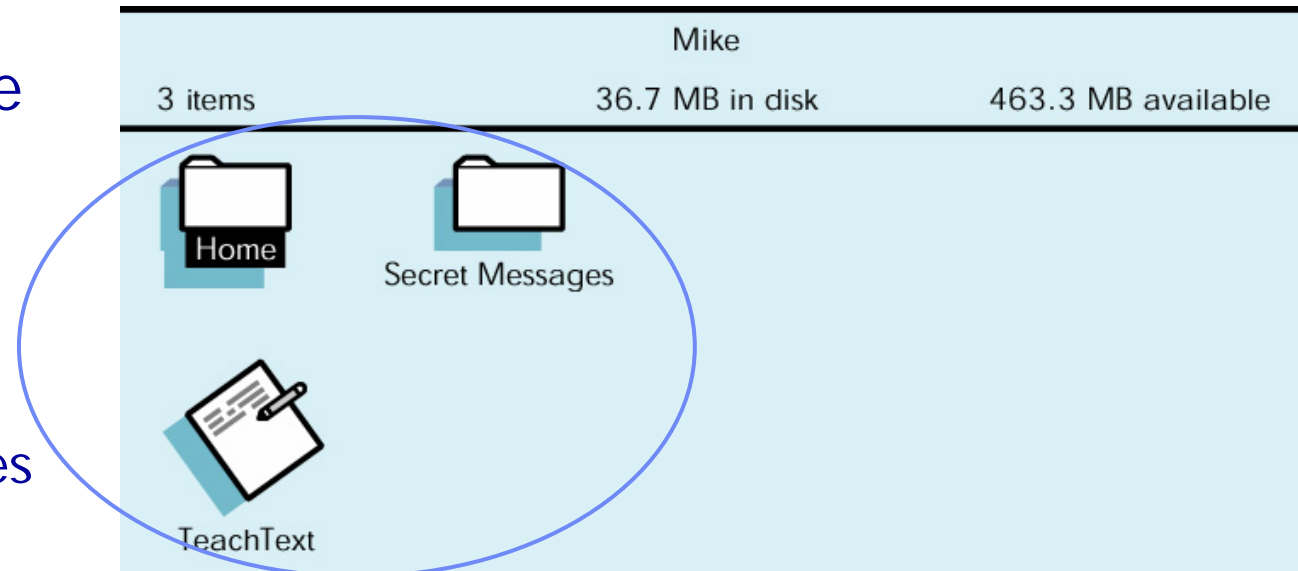
User Name:  Schneider
Password:  XXXXX   (Blocked out for security reasons)
$          ($ is the prompt. The system is now waiting for a user request.)
    
```



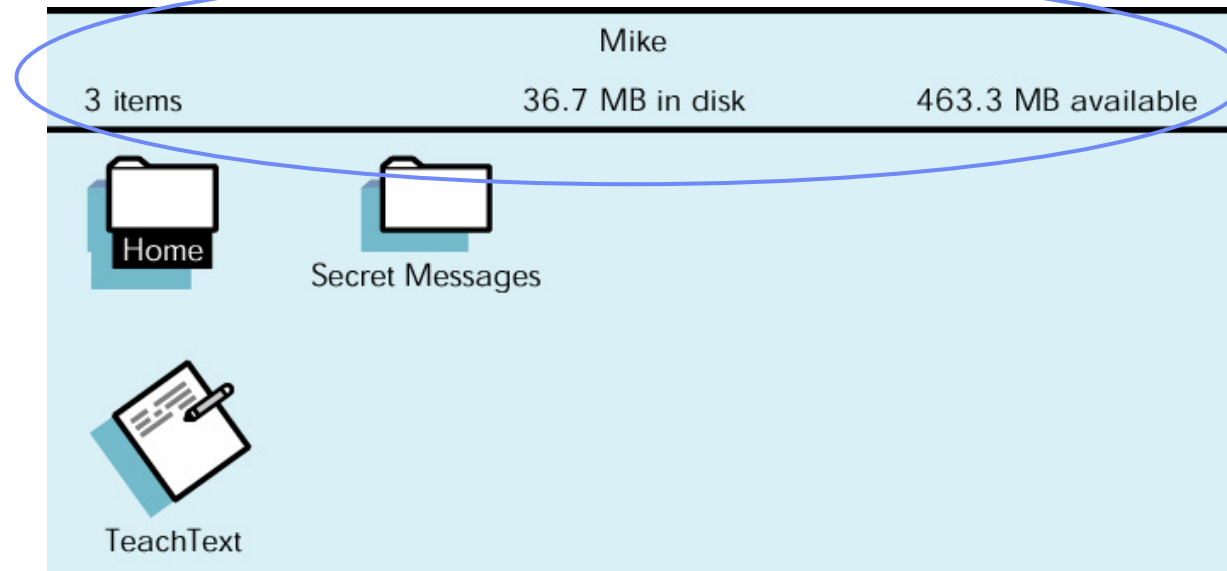
File system

➤ Presentazione dei file all'utente

- Directory (cartelle)
- Nomi dei files
- Percorsi



File system



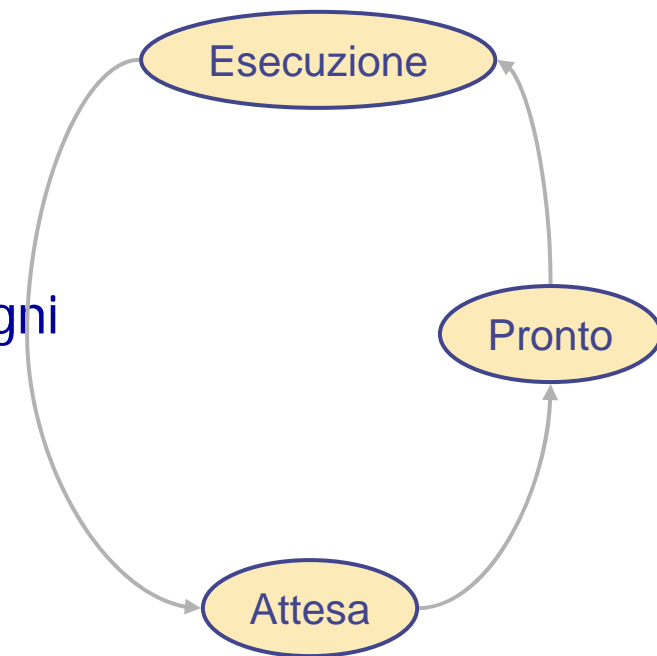
- Formattazione dei dischi
 - Dimensione
 - Nome del volume
 - Numero oggetti presenti
- FAT (File Allocation Table)
 - Contiene le corrispondenze <nome file> → <blocco di inizio>
 - Settori concatenati: <1° blocco file> → <2° blocco file> → ...

Protezione e sicurezza

- Controllo di accesso al sistema
 - Giorni dorati (anni '50 e '60) si parlava di sicurezza fisica (Guardie, porte e sbarramenti)
 - Avvento reti telecomunicazione (anni '70) compito demandato al sistema operativo
- Autenticazione utente
 - User name e password
 - File delle password crittografato
- Crittografia
 - "ABC" → 01000001 01000010 01000011
 01000001 01000010 01000011 << 6 + 15
 → 01010000 10010000 11001111 → "P © O"
- Autorizzazione operazioni
 - Lista di autorizzazioni crittografata
 - Lettura, aggiunta, modifica, cancellare, eseguire

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
- stato di esecuzione
 - 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

Programma A

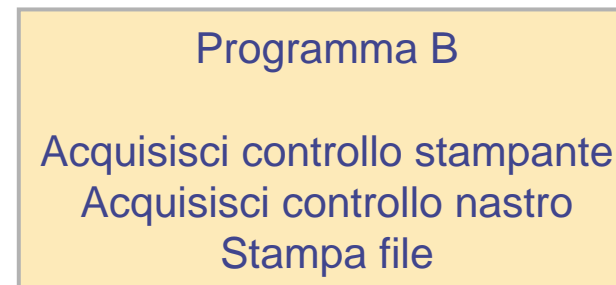
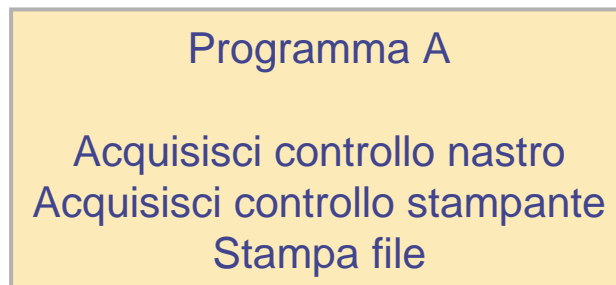
Acquisisci controllo nastro
Acquisisci controllo stampante
Stampa file

Programma B

Acquisisci controllo stampante
Acquisisci controllo nastro
Stampa file

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

Utilizzo sicuro delle risorse

- Formalmente, lo stato di deadlock si raggiunge se una serie di programmi sono in stato di attesa che un evento accada, e tale evento può essere generato soltanto da uno degli altri programmi in attesa
- Algoritmi di prevenzione (o ripresa) da deadlock
 - Se un programma non possiede tutte le risorse necessarie alla sua esecuzione, allora deve restituire quelle in suo possesso e ripetere la richiesta di allocazione

Attesa	Pronto	Esecuzione
A		
B		

Prevenzione del deadlock

Richiesta	Risposta SO	Controllo Nastro	Controllo Stampante
A: Nastro, Stampante	Granted	Locked.A	Locked.A

- A richiede le risorse
- SO assegna
- A passa in attesa che il dato arrivi dal nastro

Attesa	Pronto	Esecuzione
	B	A

Prevenzione del deadlock

Richiesta	Risposta SO	Controllo Nastro	Controllo Stampante
B: Nastro, Stampante	Denied	Locked.A	Locked.A
A: rilascia nastro			Locked.A

- B richiede le risorse
- SO nega
- B passa in attesa di potere ripetere la richiesta
- A passa in pronto e rilascia il nastro

Attesa	Pronto	Esecuzione
A		B

Prevenzione del deadlock

Richiesta	Risposta SO	Controllo Nastro	Controllo Stampante
			Locked.A

- A passa in esecuzione per eseguire stampa
- B passa in pronto

Attesa	Pronto	Esecuzione
B	A	

Prevenzione del deadlock

Richiesta	Risposta SO	Controllo Nastro	Controllo Stampante
			Locked.A

- A esegue stampa, e passa in attesa stampa completa
- B passa in esecuzione per reiterare la richiesta

Attesa	Pronto	Esecuzione
	B	A

Prevenzione del deadlock

Richiesta	Risposta SO	Controllo Nastro	Controllo Stampante
A: rilascia stampante			
B: Nastro, Stampante	Granted	Locked.B	Locked.B

- A finisce stampa, rilascia stampante e passa in pronto
- B richiede risorse
- SO assegna
- B passa in attesa dato sia caricato dal nastro
- ...

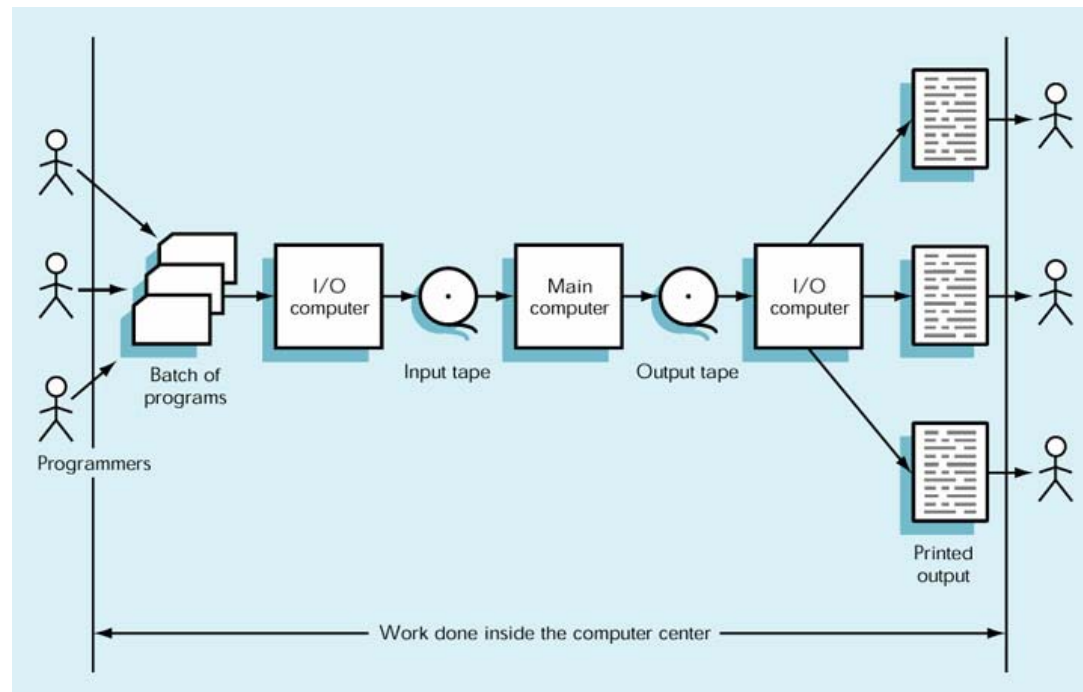
Attesa	Pronto	Esecuzione
	A	B

Panoramica storica

- Prima generazione (1945-1955)
 - Assemblatori e caricatori
 - Nessun sistema operativo
 - Operazioni manuali
- Prenotazione delle macchine a tempo, ma per la maggior parte del tempo la macchina rimaneva *idle*

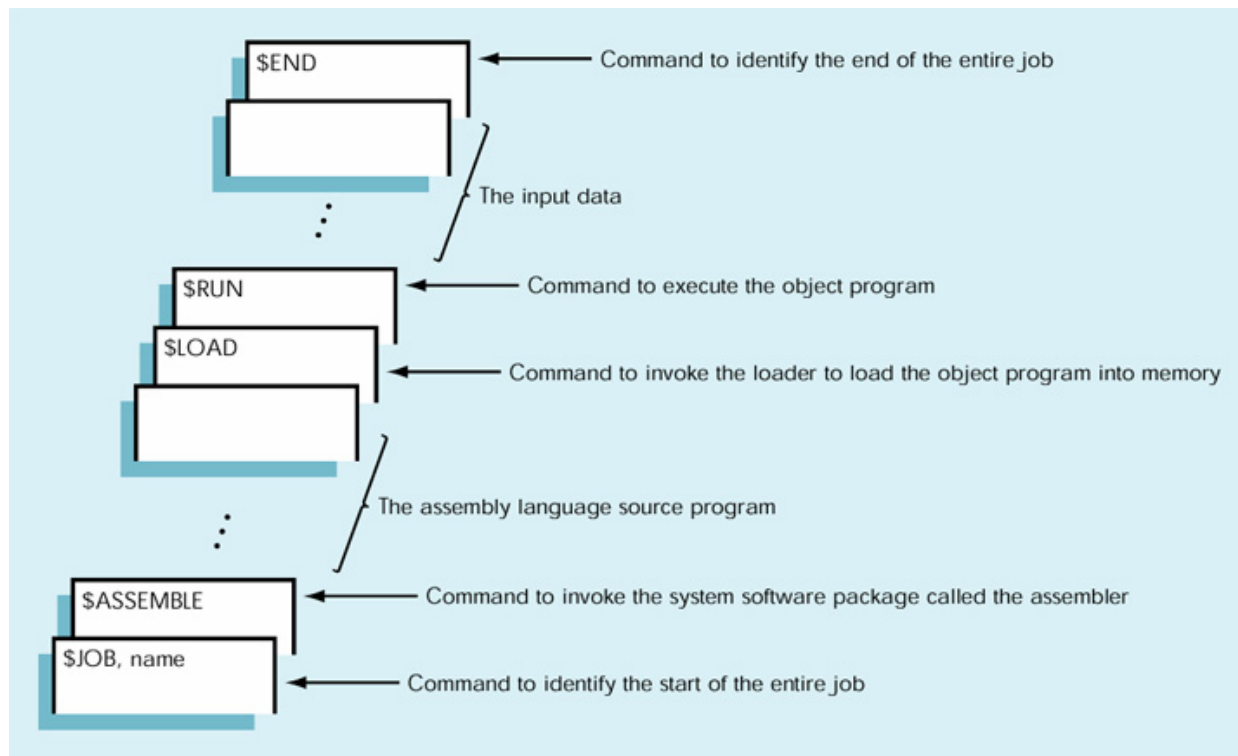
Seconda generazione (1955-1965)

- Sistemi operativi batch
 - Il programmatore consegnava agli operatori il programma
 - Più programmi sono organizzati in un lotto
 - Esecuzione del lotto
 - Restituzione dei risultati dei singoli programmi ai programmatori



Elaborazione in lotti

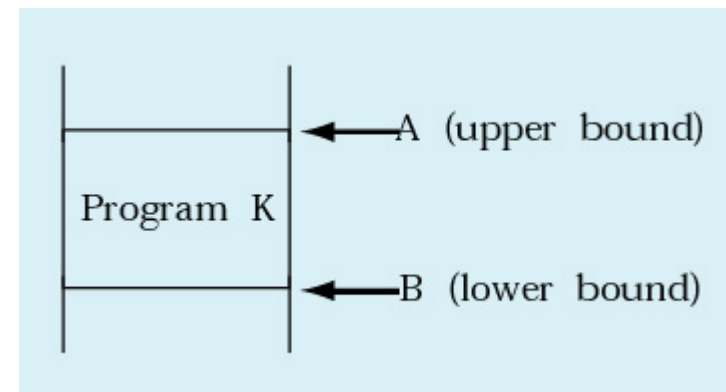
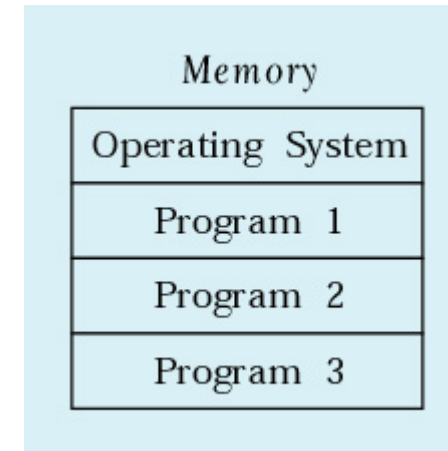
- Nascita di un linguaggio di comando, o job control language
 - Usato per inviare comandi al SO per l'esecuzione di un job
- Job
 - Miscela di programmi, dati e comandi al SO





Terza generazione (1965-1985)

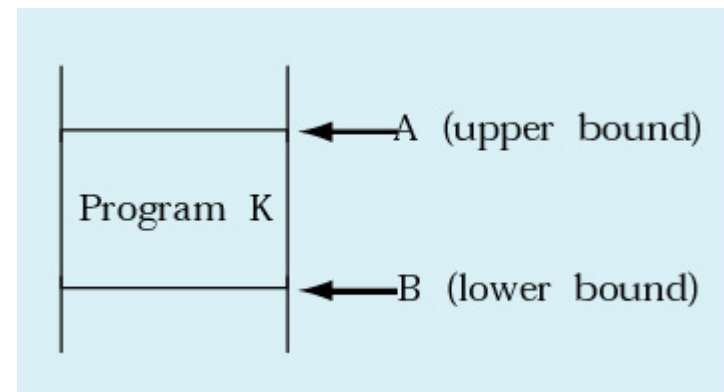
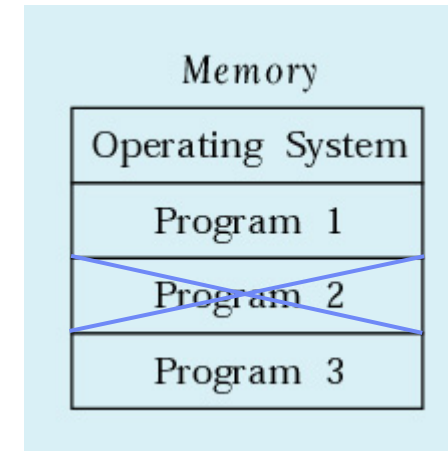
- Avvento dei circuiti integrati e aumento velocità
- Necessità di ridurre il tempo idle dovuto alla esecuzione di un solo programma alla volta
- **Sistemi operativi multiprogramma**
 - Partizionamento della memoria
 - Scheduling con ciclo esecuzione/attesa/pronto
- Nasce il rischio di interazione non voluta tra i programmi concorrentemente in memoria dovute a errori di programmazione
 - Mantenimento della lista delle locazioni di inizio e fine di ciascun programma, e dei dati ad esso necessari
- Suddivisione delle istruzioni in due classi
 - istruzioni utente (qualunque programma)
 - istruzioni privilegiate (solo SO)



Terza generazione (1965-1985)

➤ Sistemi operativi multiprogramma – gestore della memoria

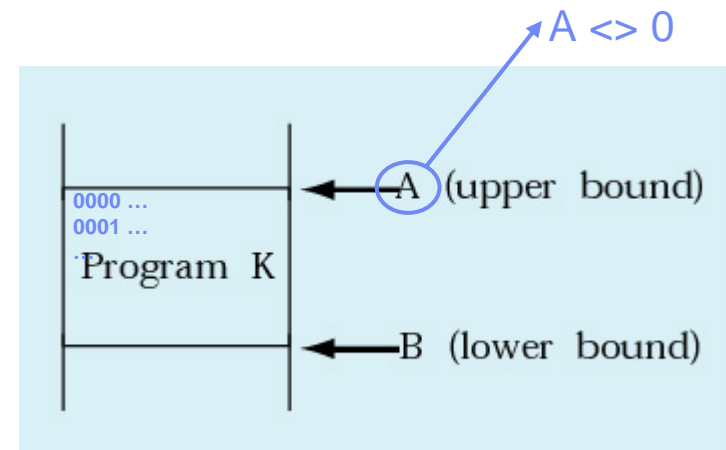
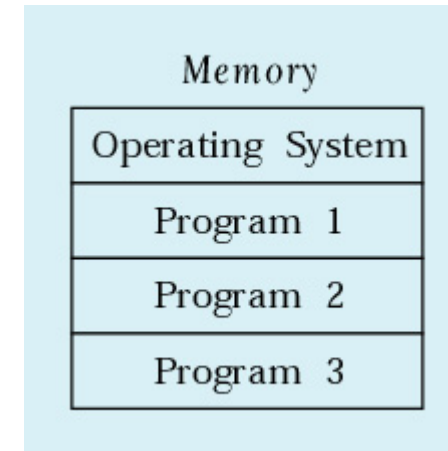
- Che fare quando un programma termina e rilascia la sua porzione di memoria?
 - ◆ Inserire un nuovo programma al posto del vecchio? E se non ci entra?
 - ◆ Lasciar perdere?



Terza generazione (1965-1985)

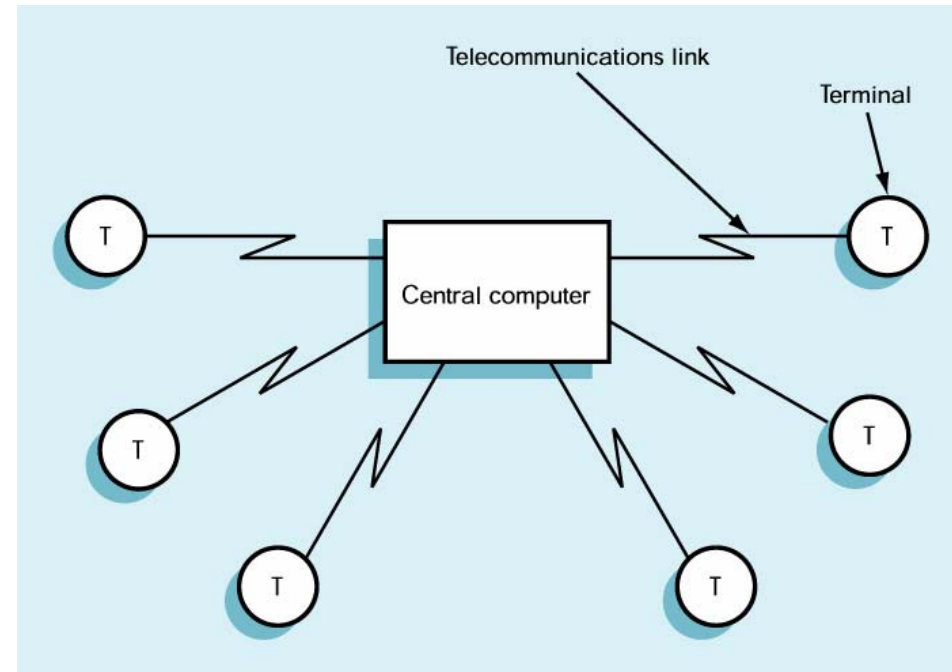
➤ Sistemi operativi multiprogramma – gestore della memoria

- Ma se un programma oggetto inizia da 0 come faccio a garantire di poterlo caricare sempre all'indirizzo 0?
 - ◆ Uso il punto di inizio per caricare correttamente il PC.
 - ◆ Codice rilocabile



Terza generazione (1965-1985)

- Sviluppo dei sistemi di telecomunicazione e delle reti di calcolatori
- Nascono i sistemi multiterminale
- Sistemi time-sharing
 - Il calcolatore centrale serve contemporaneamente diversi Utenti/Terminali, offrendo "fette di tempo macchina" a turno
 - Programmi compute-bound
 - Esecuzione in time-slice
- Un programma può usare tempo macchina all'interno di una time-slice o fino alla esecuzione di I/O





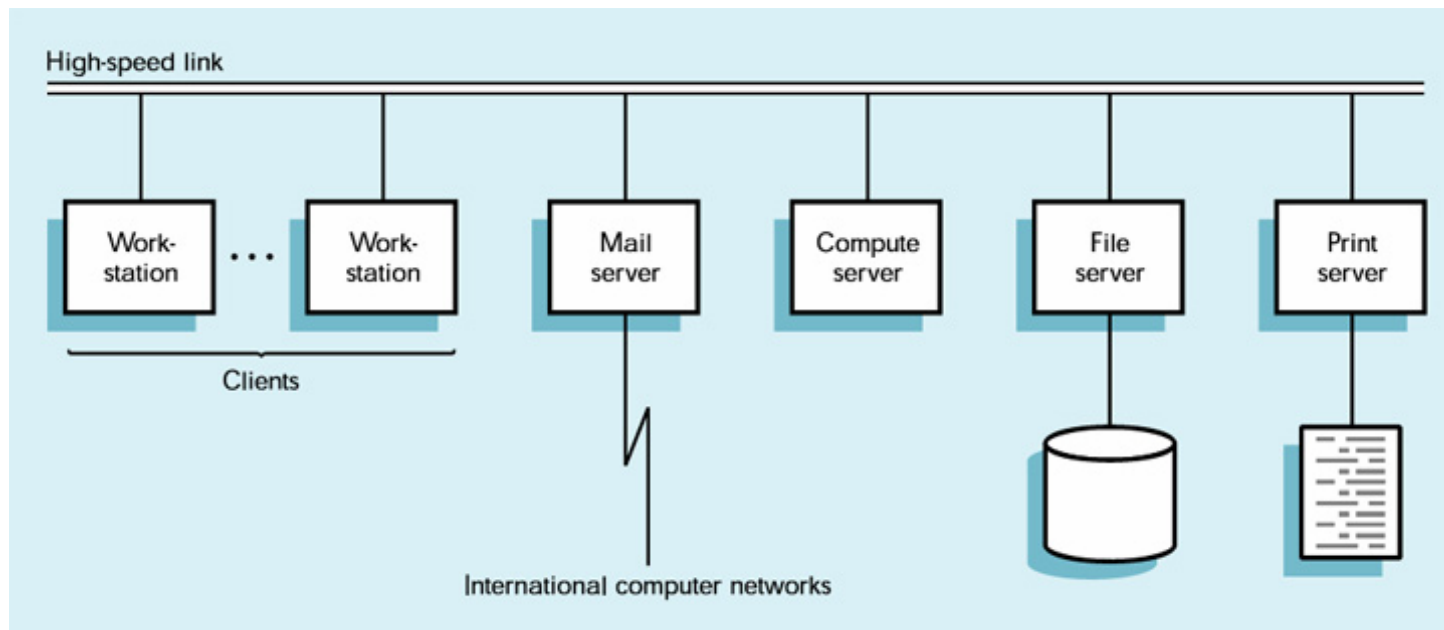
Quarta generazione (1985-oggi)

- Nascono i sistemi di elaborazione per singolo utente
- Necessità di un ambiente virtuale per
 - Elaborazione locale
 - Accesso remoto ad altri utenti e risorse condivise
- Sistemi operativi con GUI
- Avvento dei sistemi operativi in rete e delle reti locali (LAN – Local Area Network)



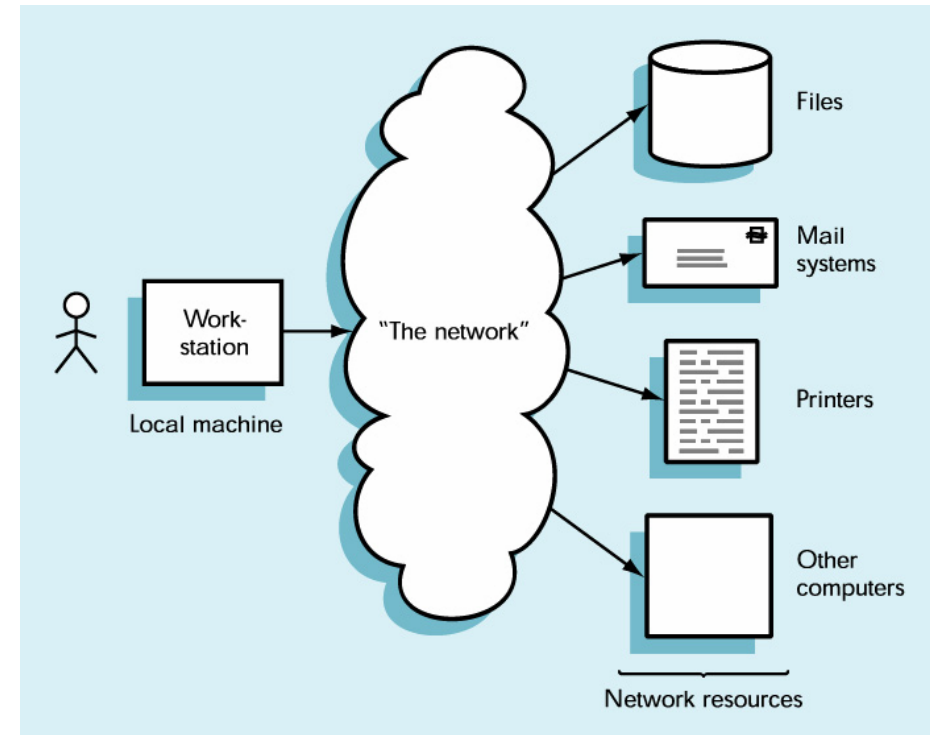
LAN – Local Area Network

- Si tratta di un'area di estensione limitata nella quale sono distribuiti un insieme di componenti, connessi attraverso un bus ad alta velocità
 - Stazioni di lavoro (PC client)
 - Server di posta
 - File server per la gestione e l'archiviazione dei dati e dei programmi
 - Print server per la condivisione di stampanti ad alte prestazioni tra più utenti
 - Compute server per l'esecuzione di programmi compute-bound



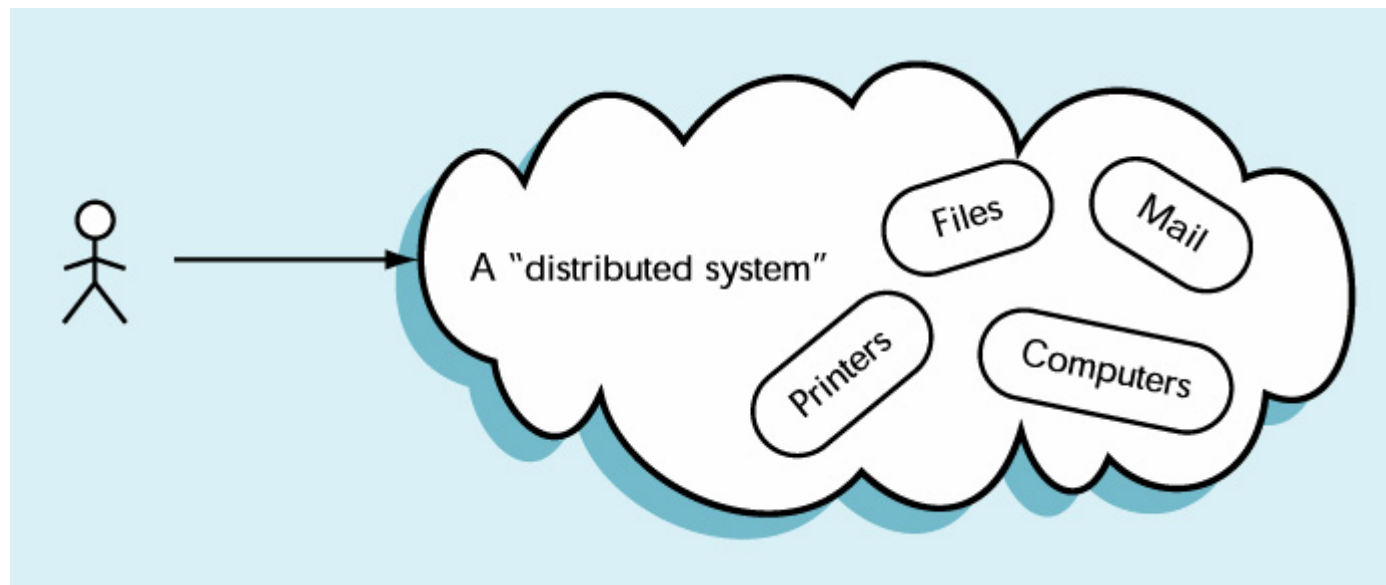
Ambiente virtuale

- File server
 - Dischi di grandi dimensioni
 - Sistemi di backup
- Server di stampa
 - Possibilità di condividere stampanti di alta qualità tra più utenti
 - Mantenere alto il livello di utilizzazione
- Server di elaborazione
 - Utilizzato per simulazioni e calcoli pesanti
- Server di posta
 - Consente la gestione dei messaggi email, sia interni che da/verso l'esterno
 - Implementa politiche di sicurezza e filtraggio



Quinta generazione (??)

- Sistemi embedded (annegati)
- Sistemi operativi in tempo reale
- Sistemi operativi paralleli
- Sistemi operativi distribuiti





Panoramica storica

Generazione	Date	Caratteristiche
Prima	1945-1955	Nessun Sistema operativo Programmi direttamente costruiti sulle macchine es 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