

Sistemi Operativi

Presentazione del corso (A.A. 2010-2011 - 9 CFU)

prof. Marco Lapegna

- tel. 081 675623
- studio 155 DMA (VI liv.)
- <http://wpage.unina.it/lapegna>

Lezione 1:

INTRODUZIONE AI SISTEMI OPERATIVI

- cos'è un sistema operativo
- evoluzione dei sistemi operativi
- scopi dei sistemi operativi
- funzioni dei sistemi operativi

Obiettivo del corso

Il corso intende fornire
**una introduzione alla struttura e alle funzioni
dei moderni sistemi operativi,**
esaminando

- le componenti fondamentali
- le metodologie di progettazione,
- gli algoritmi e le strutture dati utilizzate
- gli strumenti di base

Ma..... cos'è un Sistema Operativo?

La risposta in una esigenza

Chi ha mai usato uno strumento elettronico **facendo
riferimento direttamente alle sue componenti
elettroniche?**



Quindi, in generale

UN
SISTEMA OPERATIVO
E'

un **ambiente software** che agisce da **intermediario** tra **l'utente** e **l'hardware** di un computer.



Dal punto di vista dell'utente, un S.O.

- Permette **l'esecuzione dei programmi** e rende più semplice la soluzione di possibili problemi legati alla gestione della macchina
- Rende il sistema di calcolo **semplice da usare**.



E' un ambiente per eseguire programmi in modo facile ed efficiente.

Dal punto di vista dell'hardware, un S.O.

- **Gestisce le risorse**: controlla ed alloca le risorse hardware (in modo equo ed efficiente).
- **Controlla l'esecuzione** dei programmi utente e le operazioni sui dispositivi di I/O facendo fronte ad eventuali errori

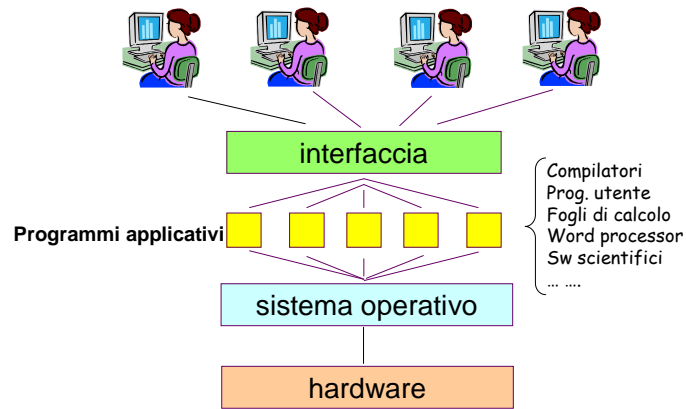


E' un insieme di programmi sempre in esecuzione e a diretto contatto con l'hardware (Kernel)

Componenti di un sistema di calcolo

1. **Hardware** — fornisce le risorse fondamentali di calcolo (CPU, memoria, device di I/O).
2. **Sistema Operativo** — controlla e coordina l'utilizzo delle risorse hardware da parte dei programmi applicativi dell'utente.
3. **Programmi Applicativi** — definiscono le modalità di utilizzo delle risorse del sistema, per risolvere i problemi di calcolo degli utenti (compilatori, database, video game, programmi gestionali).
4. **Interfaccia** — permette agli utenti di accedere alle risorse del sistema attraverso un linguaggio di comando, desktop grafici o altro
5. **Utenti** — persone, altri macchinari, altri elaboratori.

Componenti di un sistema di calcolo

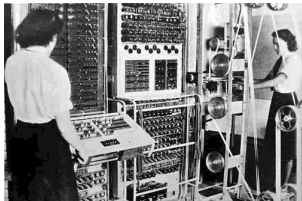


Quali sono le funzioni di un S.O. ?

La struttura dei moderni sistemi operativi e' il frutto di aggiunte di programmi introdotti nel tempo per far fronte a specifiche esigenze

Uno sguardo alla storia del calcolo ...

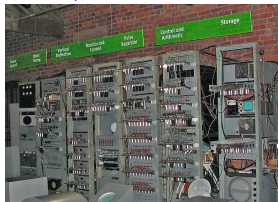
Fino al 1948-49



Colossus (Inghilterra, 1944): usato per decriptare codici durante la guerra. **Elettronico**, programma non in memoria (cavi e spinotti)

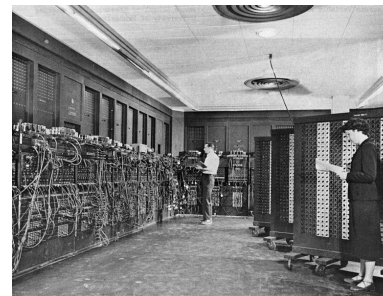


Zuse Z3 (Germania, 1941). Primo calcolatore elettromeccanico, programma in memoria



SSEM (Inghilterra, 1948). Non un calcolatore, ma un dispositivo per sperimentare nuove tecnologie per le memorie (**elettronico e programma in memoria**)

ENIAC (Pennsylvania- USA 1948)



Primo calcolatore **elettronico con programma in memoria**

Ispirato dai lavori di John Von Neumann

La memorizzazione del programma avveniva mediante lettura di nastri perforati

Macchina di Von Neumann: calcolatore con CPU separata dalla memoria che e' capace di **contenere istruzioni e dati**



Anni 1940-1950

- interazione diretta uomo - calcolatore
- assenza di qualunque tipo di software di sistema

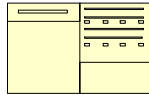


Svantaggi:

- Elaborazione molto lenta e inefficiente
- Alta possibilità di errori
- gestione inefficiente del sistema



Soluzione
Schede e nastri magnetici



Un centro di calcolo negli anni 50



Un lettore di nastri



IBM 701

Alcuni problemi

- La nascita di **numerosi strumenti di I/O**, ognuno con **caratteristiche differenti**
(buffer, flag, registri, bit di controllo, bit di stato,...)



Una semplice operazione di I/O poteva essere composta di numerose istruzioni di linguaggio macchina (magari da ripetere molte volte da molti utenti)



Scrivere una procedura specifica per ogni dispositivo (Driver del dispositivo)

Nascita delle librerie di I/O

Primi linguaggi di programmazione

- Fortran (1957), Algol (1960) e Cobol (1961)
- Necessita' di un traduttore (**compilatore**)
- Passi da eseguire:
 1. Montare il nastro del compilatore
 2. Tradurre il programma utente
 3. Smontare il nastro del compilatore
 4. Collegare il programma utente tradotto con le librerie di I/O
 5. Montare il nastro con il programma utente
 6. Esecuzione del programma

Ancora troppi passi (tutti manuali !!!)

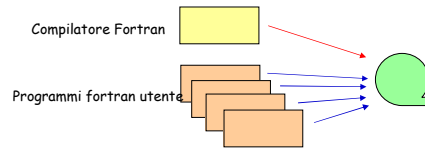
Sistemi batch

Programmi con **caratteristiche simili** possono essere raggruppati in lotti (**batch**)



Si riduce il tempo di preparazione

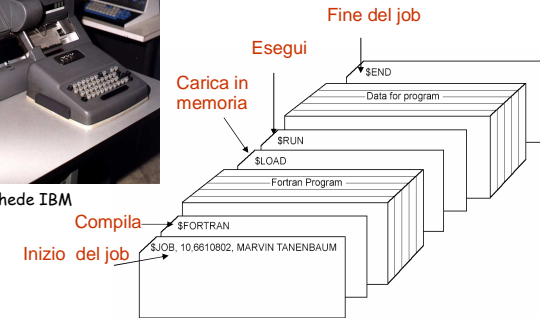
Esempio : un insieme di programmi Fortran



Struttura di un tipico job Fortran

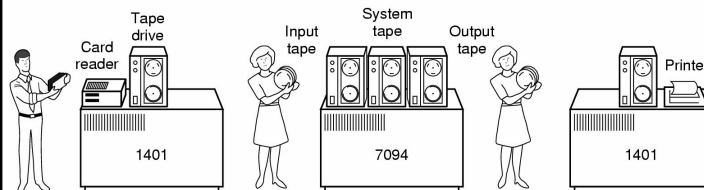


Perforatrice di schede IBM



La struttura del job e' definita dalle **schede di comando**

Sistemi batch (a lotti)

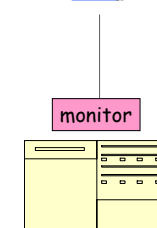


- Un lotto di job sotto forma di schede viene **trascritto su di un nastro**
- Il nastro viene montato nel **sistema centrale (7094)** che li elabora **uno alla volta**. I risultati sono trascritti su di un ulteriore nastro (**nel frattempo e' possibile cominciare a creare un altro nastro di job sul sistema ausiliario 1401**)
- Il nastro con i risultati viene montato **sul sistema ausiliario 1401** che **li stampa** (nel frattempo il sistema centrale 7094 elabora altri job su un altro nastro)

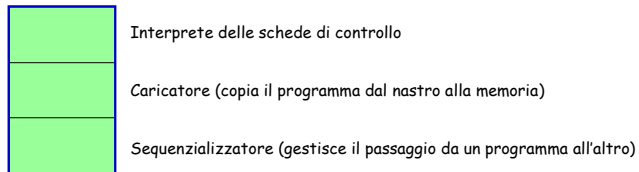
Monitor residente

Con i sistemi batch il controllo viene trasferito automaticamente da un job al successivo da un **programma residente in memoria** chiamato **monitor** che **legge le schede di comando** e chiama le opportune routine.

- Monitor residente:
 - Legge le schede di controllo
 - Chiama le relative routine di servizio
 - Usa il programma utente e i dati come input di tali routine



Il monitor residente



Il monitor garantisce le **sequenza automatica dei programmi** così come era indicato nelle schede di controllo



La **lentezza dell'intervento umano** viene **sostituita** dal calcolatore stesso

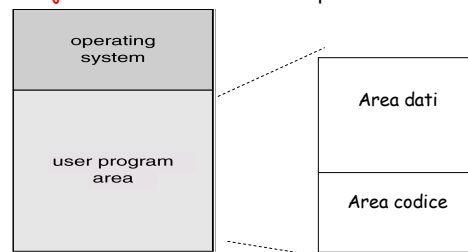
Primi sistemi operativi per sistemi batch

- **GM OS** (1955)
 - Per alcuni è il primo sistema operativo in assoluto per sistemi batch. Sviluppato dalla General Motors per l'IBM701, basato su monitor residente
- **SAGE** (1957 ?)
 - Semi-Automatic Ground Environment system. Programma di controllo per sistemi IBM. Primo sistema operativo real time sviluppato in ambito militare.
- **Fortran Monitor System** (1958)
 - Sistema operativo sviluppato dall'aviazione americana per calcolatori IBM. Primo con supporto per un linguaggio ad alto livello.
- **SOS** (1959)
 - Sistema operativo sviluppato dall' IBM SHARE Users Group per l'IBM 709

Sistemi batch: caratteristiche

- Presuppongono un operatore ≠ utente
- Assenza di interazione fra utente e job a *run-time*.
- Presuppongono come periferica di ingresso un lettore di schede o nastri.
- Riducono il tempo di setup riunendo in lotti (*batch*) job simili.
- Aumento del throughput
- viene eseguito **un solo job** alla volta fino al suo completamento

Uso della memoria in un sistema batch negli anni 50



Sistemi batch : problemi

- le operazioni di elaborazione e di I/O non possono essere svolte contemporaneamente.
- E' possibile tenere in memoria un solo job alla volta
- Lentezza dei lettori di schede e nastri rispetto alla CPU (anche 3 ordini di grandezza).



Uso inefficiente della CPU

Soluzione: uso di memorie di massa veloci ad **accesso diretto (dischi)**

Un centro di calcolo negli anni '60



Lettori dei dischi !!

IBM 360, 1964

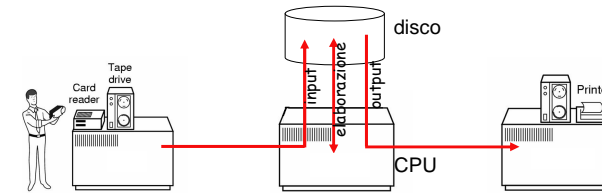
OS/360 primo s.o. portabile su una famiglia di calcolatori

Marco Lapegna - Sistemi Operativi

25

Lezione 1: introduzione

Spooling



- I job e i dati vengono caricati **automaticamente** dal lettore di schede al disco
 - Mentre viene eseguito un job, il SO...
 - Legge il prossimo job dal lettore di schede su un'area del disco (job queue).
 - Stampa l'output di job eseguiti **precedentemente**, copiandoli dal disco su un nastro.
- Sovrapposizione di I/O e elaborazione**

Marco Lapegna - Sistemi Operativi

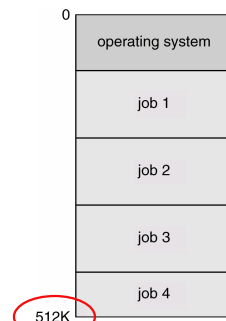
26

Lezione 1: introduzione

Sistemi multiprogrammati

Più job vengono mantenuti nella memoria principale **contemporaneamente** in particolari strutture (job pool) e l'uso della CPU viene diviso fra loro

(Scheduling della CPU)



Uso della memoria in un sistema multiprogrammato negli anni 60

Marco Lapegna - Sistemi Operativi

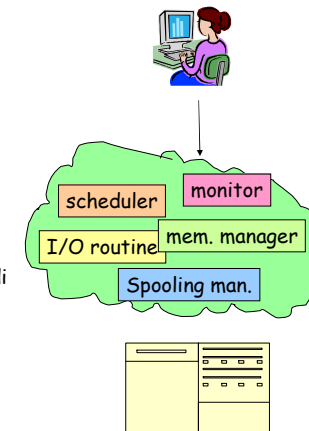
27

Lezione 1: introduzione

S.O. per calcolatori con multiprogrammazione

- la multiprogrammazione impone al SO...

- Presenza di routine per I/O.
- Gestione della memoria
- Scheduling della CPU —
- Gestione dei dischi e delle unità di I/O.



Marco Lapegna - Sistemi Operativi

28

Lezione 1: introduzione

Sistemi multiprogrammati: problemi

Se nel job pool e' presente un **job molto lungo**, eventuali **job piu' piccoli devono attendere** la fine di tale job anche se devono usare la CPU per poco tempo



tempo medio di elaborazione elevato

Soluzione: **sospendere l'esecuzione** di job che superano un fissato tempo limite (time slice)

Sistemi time-sharing

- un **job viene sospeso** quando
 - deve effettuare una **operazione di I/O**
 - ha **esaurito il suo tempo limite**
- La CPU viene commutata tra più job che vengono mantenuti contemporaneamente in memoria e sul disco
- I job sono sottoposti a **swap-in** dal disco alla memoria ed a **swap-out** dalla memoria al disco.



Riduzione del tempo medio di attesa

Sistemi interattivi

Con la **multiprogrammazione** e il **time sharing** si riducono i tempi medi di attesa e ogni utente ha la sensazione di **essere l'unico utente** del sistema



SISTEMI INTERATTIVI

- permettono la **comunicazione on-line tra utente e sistema**; quando il SO termina l'esecuzione di un comando, si aspetta il successivo comando da tastiera.
- devono essere sempre **disponibili per l'accesso a dati e codice** da parte degli utenti (**File system on-line**).

problemi

Il **time sharing** permette l'accesso alla cpu a **molti programmi residenti in memoria**

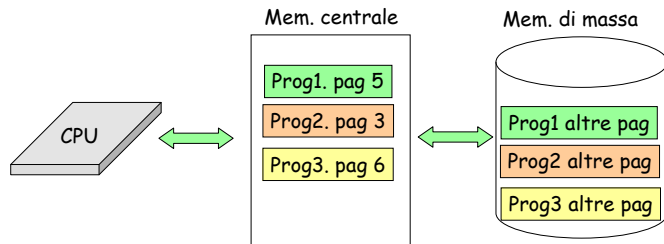
La memoria ha una capacita' limitata

Come fare a tenere numerosi programmi (anche grandi) **contemporaneamente in memoria?**

soluzione

- Dividere il programma in pezzi (pagine) e conservare:
- in memoria centrale solo la pagine con la sezione di codice da eseguire
 - in memoria di massa il resto delle pagine

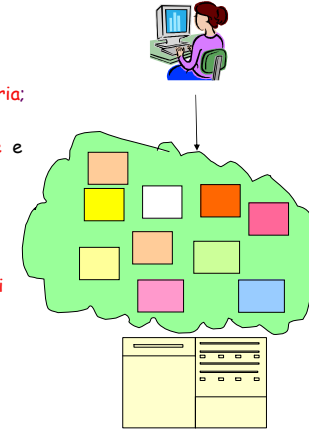
MEMORIA VIRTUALE



S.O. per calcolatori con time-sharing

- Il time-sharing impone al S.O...

- La gestione e la protezione della memoria;
- La gestione della memoria virtuale;
- La gestione di un file system on-line e della memoria secondaria di supporto;
- La presenza di meccanismi per l'esecuzione concorrente, la comunicazione e la sincronizzazione dei job;
- La presenza di meccanismi per evitare i deadlock.
- La presenza di differenti unità di I/O



Sistemi operativi negli anni '60

- **Atlas (Inghilterra, 1961)**
 - gestione particolarmente efficiente delle memorie e uso della memoria virtuale
- **CTSS (California, 1961)**
 - Il "bisnonno di Linux"
- **OS360 (USA 1965)**
 - Primo s.o. portabile su una famiglia di calcolatori IBM
- **XDS-940 (California, 1965)**
 - Tra i primi ad utilizzare il time sharing e introduzione della duplice modalità di esecuzione (utente / sistema)
- **THE (Olanda, 1968)**
 - Progettazione accurata e gestione efficiente e semplice dei programmi in esecuzione con i semafori
- **RC4000 (Danimarca, 1970)**
 - Introduzione di moderni meccanismi di comunicazione tra i programmi in esecuzione

Storia di Unix (1)



John McCarthy e **Herb Teager** (Stanford, 1961) pongono le basi per il primo S.O. time sharing: il **CTSS** (Compatible Time Sharing System) Sistema prototipale, piccolo e con poche funzionalità



Fernando José Corbató (MIT, 1965), tra i progettisti di CTSS dirige il progetto **MAC** per il S.O. **MULTICS** (MULTIplexed Information and Computing Service) Evoluzione del CTSS, ma grande, complesso e poco efficiente



Kenneth Thompson (Bell labs, 1969), dopo il ritiro della Bell Labs dal progetto **MULTICS**, sviluppa **UNIX** semplificando il progetto **MULTICS**. S.O. con time sharing e memoria virtuale. Disponibile per calcolatori medio/grandi.



Storia di Unix (2)



Dennis Ritchie (Bell Labs, 1972), sviluppa il linguaggio C e riscrive UNIX nel nuovo linguaggio. Primo S.O. portabile su differenti piattaforme



Bill Joy (Berkeley, 1978), a partire dal codice che i Bell Lab furono tenuti a distribuire a causa di leggi antitrust, guida un gruppo di ricercatori nello sviluppo di BSD Unix con funzionalità di rete TCP/IP. Una delle più diffuse distribuzioni negli anni 80.



Paul Allen e Bill Gates (1980), alla Microsoft sviluppano XENIX, prima versione commerciale per microcomputer. Seguiranno le prime versioni commerciali per processori Intel: Venix, QNX, Idris... Poco diffuse per la scarsa potenza dei processori per PC

Storia di Unix (3)

Richard Stallman (Stanford, 1985), lancia il progetto GNU per la distribuzione libera e gratuita del software di Unix. Sviluppa solo alcuni tool (compilatori, debugger, ..)

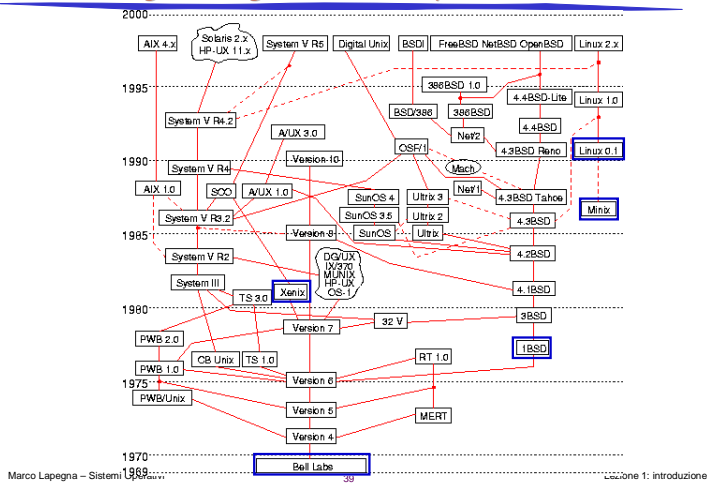


Andrew Tanenbaum (Amsterdam, 1986), professore alla Vrije University sviluppa Minix, piccolo S.O. Unix per processori Intel (ora sufficientemente potenti). Sviluppato per usi didattici - gratuito



Linus Torvalds (Finlandia, 1992) sviluppa Linux a partire da Minix e dai tools realizzati nel progetto GNU. Free e open source. Motivo: costi troppo alti per le licenze. Versione per processori Intel poi adottata da tutte le grandi industrie

L'albero genealogico di Unix (parziale)



I Personal Computer

- Avvento dei microprocessori negli anni 70
- miglioramento dei processi tecnologici

Obiettivo:

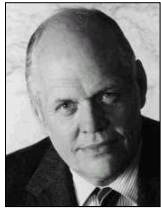
Costruire calcolatori economici e facili da usare (personal computer)



Federico Faggin (1971)

- Introduzione di numerosi mezzi di I/O (tastiera, mouse, monitor, piccole stampanti).
- Possono adottare tecnologie già sviluppate per i sistemi operativi per mainframe;
- un solo utente utilizza il PC e quindi non sono necessarie tecniche sofisticate per l'utilizzo della CPU, né sono richieste funzioni avanzate di protezione.

Chi ha avuto ragione?

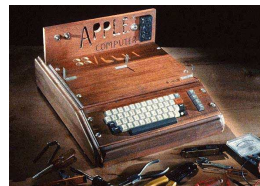


"There is no reason anyone would want a computer in their home." -

Ken Olsen, presidente, e fondatore della Digital Equipment Corp., 1977



Steven Wozniak, Steve Jobs



L'Apple I (1977)

S.O. per personal computer

- MS-DOS (MicroSoft Disk Operating System -1980)
- Windows (1985), Windows 95, Windows 98, Windows Me
- Windows NT, Windows 2000, Windows XP



- MacOS



Mac OS

- Minix (1987)
- Linux (1994)

Caratteristiche:

- facilita' di uso mediante GUI
- efficienti per elab. testi, fogli elettronici,...
- piccole dimensioni



Storia di Windows (1)

1980 - Microsoft, dietro commessa della IBM produce un S.O. per personal computer chiamato PC-DOS 1.0. Per i PC IBM compatibili viene prodotta una versione simile chiamata **MS-DOS**

1982 - DOS V.1.2 con driver per **floppy disk**

1983 - DOS V.2.0 con driver per **hard disk** e gestione del **file system**

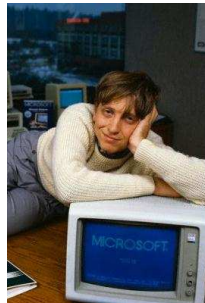
1984 - DOS V.3.0 con supporto di **rete**

1985 - Windows 1.0 prima **interfaccia grafica**

1987 - Windows 2.0 con funzionalita' di **multitasking**

1988 - DOS V.4.0 con shell e gestione della **memoria estesa**

1991 - DOS V.5.0 gestione piu' **efficiente della memoria**



B. Gates - 1977



Storia di Windows (2)

1992 - **Windows 3.1.** oltre 3 milioni di copie in due mesi

1993-94 - Windows NT 3 e NT4. **S.O. object oriented** per server di alto livello. Stessa interfaccia grafica di Windows 3.1 ma kernel completamente riscritto

1995 - Windows 95: S.O. per **applicazioni a 32 bit.** Non piu' una interfaccia grafica come Windows 3.1 ma un completo S.O.

1996-1998 - Windows CE 3.0. S.O. per **computer palmari**

1998 - Windows 98. **browser** integrato, supporti **Java** e **HTML** e vari supporti hardware (USB, firewire, DVD,...)

2000 - Windows 2000 con supporto molto efficiente per le **connessioni di rete**

2001 - Windows XP: supporto ai **multiprocessori**

2009 - Windows 7: supporto **multitouch** e per **CPU multicore**



B. Gates - 2004

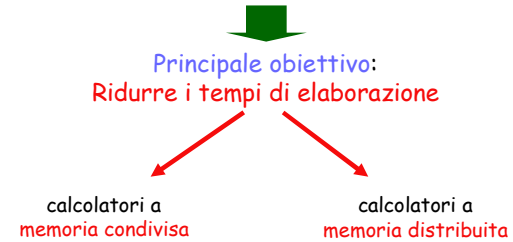
L'Unione fa la forza



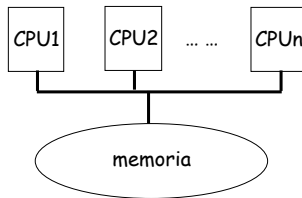
Problemi grandi ↔ Utilizzo di molte risorse

calcolatori paralleli (Tightly coupled system)

Un sistema di unità processanti **omogenee**, **strettamente collegate** che **comunicano** per risolvere problemi **su larga scala** in maniera efficiente



calcolatori paralleli a memoria condivisa



Le CPU **condividono** la **memoria** e il **clock**

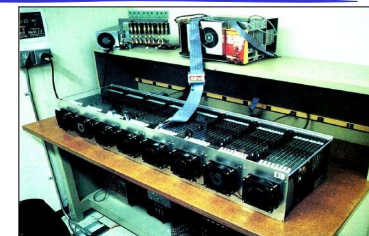
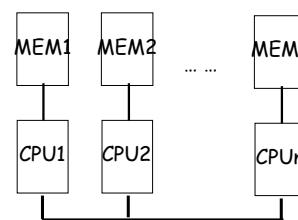
Principali svantaggi:

- sincronizzazione degli accessi alla memoria
- scarsa scalabilità



Seymour Cray e il Cray X-MP (1984)

Calcolatori paralleli a memoria distribuita



Cosmic Cube con 64 schede Intel 8086 + memoria (1983)

Ogni CPU ha una propria memoria e comunica **mediante una rete**

Principali svantaggi:

- reti lente
- comunicazione tra le CPU

Goeffrey Fox



Le reti geografiche

- 1968 primi progetti di una **rete di computer** dell'ARPA per connettere 12 università e centri di ricerca. Bandwidth = 56 Kbits



Rete GARR che connette univ. e centri di ricerca italiani

1994	2Mbit/sec
1998	32Mbit/sec
2002	2.56Gbit/sec
2006	10Gbit/sec

Crescita di 1000 volte in 8 anni (x2 in 9 mesi)



Utilizzare i **calcolatori** connessi ad una **rete** come una **unica risorsa di calcolo** (calcolo distribuito)

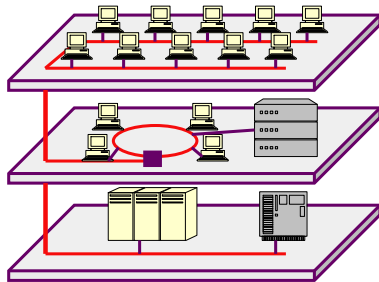
Sistemi distribuiti (*Loosely coupled system*)

Un sistema di unità processanti **non omogenee, autonome, indipendenti, geograficamente distribuite** che sono **aggregate** per risolvere problemi su larga scala in maniera efficiente



Principale obiettivo:
Aggregare risorse

Un sistema distribuito e' ...



Una **Rete Aziendale** composta da **differenti calcolatori** collegati tra loro tra **reti differenti**

- Richiedono un'**infrastruttura di rete**.
- La rete può essere una **LAN** (*Local Area Network*) o una **WAN** (*Wide Area Network*).

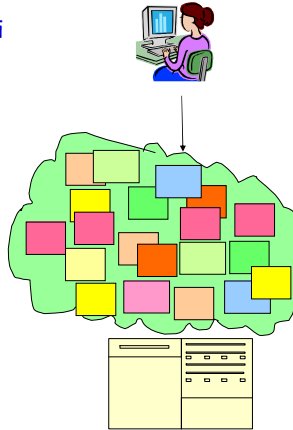
S.O. per calcolatori paralleli/distribuiti

- Sistemi operativi distribuiti
 - Minor autonomia fra computer;
 - unico sistema operativo che controlla e gestisce in maniera trasparente l'intera rete di computer.
- Sistemi operativi di rete
 - Consentono la condivisione di file;
 - Garantiscono uno schema di comunicazione;
 - Vengono eseguiti indipendentemente per ciascun computer in rete.

S.O. per calcolatori paralleli/distribuiti

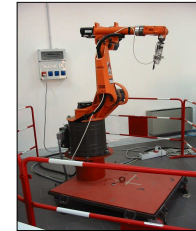
- i calcolatori paralleli/distribuiti impongono al SO...

- fault tolerance
- sincronizzazione
- gestione eterogeneità
- infrastruttura di rete
- protocolli di comunicazione
- bilanciamento del carico
- sicurezza
- Accesso a risorse remote



Sistemi operativi tempo-reale

- Spesso utilizzati per applicazioni dedicate,
 - Telecomunicazioni,
 - Difesa militare
 - Controllo traffico aereo/ferroviario,
 - Controllo di sistemi industriali



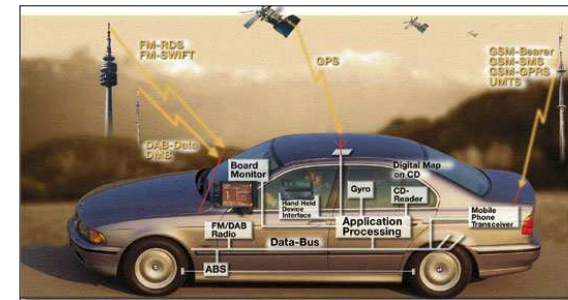
Sistemi operativi tempo-reale

Sono caratterizzati da tempi di risposta certi

- Hard real-time:
 - Memoria secondaria limitata o totalmente assente, dati memorizzati in memorie volatili o di sola lettura (ROM).
 - Non realizzano il time-sharing. Le funzionalità hard real-time non sono supportate dai SO general purpose.
- Soft real-time:
 - I task critici hanno priorità sugli altri task e la mantengono fino al completamento dell'esecuzione.
 - Utile nelle applicazioni che richiedono caratteristiche avanzate del SO (multimedia, realtà virtuale), ma non per controllo industriale e robotica.

Sistemi embedded (o integrati)

- sono i computer delle automobili, delle lavatrici, delle centraline di allarme, dei bancomat ...
- sistemi molto semplici che svolgono mansioni molto specifiche
- danno priorità alla gestione dei dispositivi fisici

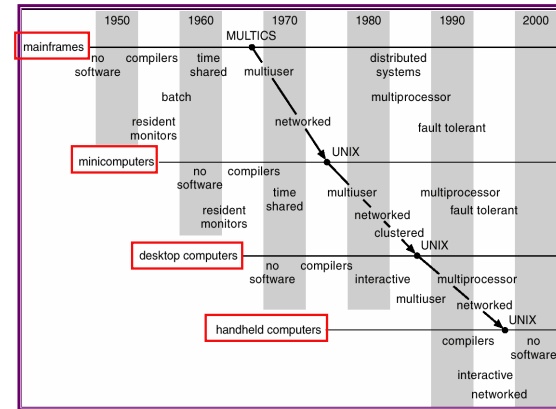


Sistemi operativi per sistemi embedded

- Caratteristiche:
 - Funzioni limitate
 - Consumo contenuto
 - Piccole dimensioni
 - Scarsa o nulla interfaccia
 - Applicazioni specifiche (agende, calendari, rubriche, pagamenti)



Migrazione temporale di concetti e caratteristiche dei SO

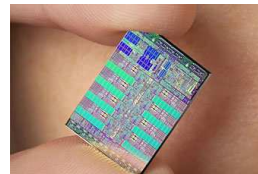


Altro esempio di "migrazione"



1984

Cray X-MP. 4 CPU



2006

Cell Processor. 9 core

Libri di testo



A. Tanenbaum
I moderni Sistemi Operativi
3 ed.
Pearson ed.- Addison Wesley



Silberschatz, Galvin, Gagne
Sistemi Operativi - VIII ed.
Pearson ed.- Addison Wesley

Libri di testo



Deitel, Deitel e Choffnes
Sistemi Operativi - 3 ed.
Pearson ed.- Addison Wesley



Ancillotti, Boari, Ciampolini, Lipari
Sistemi Operativi (2 ed.)
McGraw-Hill