



Schema della lezione

Sistema operativo

Multiprogrammazione

Precursori

Prima generazione

Seconda generazione

Terza generazione

Quarta generazione

SO'12

2.1



Schema della lezione

Sistema operativo

Multiprogrammazione

Precursori

Prima generazione

Seconda generazione

Terza generazione

Quarta generazione

SO'12

2.2

Lezione 2

Introduzione ai sistemi operativi

Sistemi operativi

13 marzo 2012

Marco Cesati

System Programming Research Group
Università degli Studi di Roma Tor Vergata

Di cosa parliamo in questa lezione?

Introduzione ai sistemi operativi e loro evoluzione

- 1 Definizione di sistema operativo
- 2 Il concetto di multiprogrammazione
- 3 I precursori dei calcolatori elettronici
- 4 I sistemi di prima generazione
- 5 I sistemi di seconda generazione
- 6 I sistemi di terza generazione
- 7 I sistemi di quarta generazione

Cosa è un sistema operativo?

I programmi di un calcolatore appartengono a due categorie: **software di base** e **software applicativo**

*Quale programmi appartenenti al software di base fanno parte del **sistema operativo**?*

- Non esiste una risposta univoca
- Perciò non esiste una definizione universalmente accettata di **sistema operativo**

La risposta dipende anche dal punto di vista:

- dell'utente finale
- del sistema hardware
- del programmatore applicativo

Il punto di vista dell'utente finale

Le varie famiglie di calcolatori rappresentano tipologie d'utilizzo differenti

Gli utenti hanno quindi diverse esigenze e priorità

- Gli utenti dei sistemi HPC privilegiano l'efficienza e la velocità d'esecuzione dei propri programmi
- Gli utenti di calcolatori aziendali, server e stazioni di lavoro favoriscono la condivisione equa delle risorse hardware
- Gli utenti dei calcolatori da tavolo e portatili danno molta importanza alla facilità d'uso ed all'aspetto delle interfacce di comunicazione
- Gli utenti di calcolatori palmari e smartphone privilegiano il risparmio di energia e interfacce semplici ed immediate
- Per gli utenti dei sistemi integrati sono cruciali responsività, tempi di bootstrap ridotti, rispetto di scadenze "real-time"

Interfacciamento verso l'utente e le sue esigenze

Il sistema operativo deve poter offrire, volta per volta, ciò che gli utenti finali realmente si aspettano



Il punto di vista del sistema hardware

Un calcolatore elettronico è costituito da centinaia di dispositivi hardware differenti

Il corretto funzionamento di ciascuno di essi necessita di un **programma di controllo** (*driver di periferica*)

Ciascuna risorsa hardware inoltre deve essere condivisa

- tra gli altri dispositivi hardware
 - esempi: bus di sistema, controllore delle interruzioni
- tra i programmi in esecuzione
 - esempi: tempo della CPU, memoria centrale
- tra i vari utenti del sistema
 - esempi: schermo, tastiera, mouse, scheda sonora

Gestore dell'hardware

Il sistema operativo svolge i ruoli di

- programma di controllo
- assegnatore di risorse

Il punto di vista del programmatore applicativo

Ciascuna periferica hardware di cui è costituito il calcolatore deve essere programmata in modo specifico

Ad esempio, sono radicalmente diverse le operazioni "hardware" da effettuare per leggere un dato da

- un disco rigido
- una memoria flash su bus USB
- un disco CD-ROM
- un nastro magnetico

Ciò che il programmatore applicativo desidera è utilizzare sempre la stessa procedura per leggere un dato da un dispositivo di memoria, in modo indipendente dalla tecnologia

Astrazione dell'hardware

Il sistema operativo fornisce al programmatore un insieme di procedure uniformi per utilizzare le periferiche hardware



Definizione di sistema operativo



Collezione di programmi di base per la gestione delle periferiche hardware del calcolatore e per la creazione di un ambiente per l'esecuzione controllata dei programmi applicativi da parte degli utenti finali

- Programma di controllo delle periferiche
- Creatore di una astrazione dei dispositivi hardware
- Assegnatore delle risorse di sistema ai programmi in esecuzione
- Ottimizzatore del sistema per soddisfare le aspettative dell'utente finale
- Realizzatore dell'interfaccia di comunicazione tra l'utente finale ed il calcolatore

Programmi e processi



Programma

Una sequenza di istruzioni macchina di un calcolatore elettronico che, nel loro insieme, svolgono un ben determinato lavoro o servizio

Processo

Una istanza di un programma in esecuzione su di un calcolatore elettronico, ovvero l'insieme di informazioni necessarie per l'esecuzione del programma stesso

*Può un processo corrispondere a due programmi? **No!***

*Può un programma corrispondere a due processi? **Sì!***

Multiprogrammazione

Esecuzione di un programma

La creazione di un processo associato al programma e l'esecuzione delle istruzioni macchina corrispondenti

Multiprogrammazione

Capacità di sospendere e riprendere l'esecuzione del flusso di istruzioni macchina di ciascun processo

In sistema con un singolo processore in ogni istante:

- si può eseguire un solo flusso di istruzioni alla volta
- un solo **processo** fa progressi verso la propria conclusione

Può un sistema monoprocesso essere multiprogrammato?

Sì, se è possibile eseguire un programma prima della conclusione di un programma avviato precedentemente

Partizione del tempo d'elaborazione

In un sistema multiprogrammato a *partizione di tempo* (o con *time sharing*) le risorse di calcolo vengono assegnate a turno ai processi attivi (associati ai programmi avviati ed eseguibili)

In un sistema con **time sharing** si assegna ciclicamente la CPU a tutti i **processi** attivi con una frequenza tale da dare all'utente l'illusione che ciascuno di essi sia eseguito senza interruzioni

Sistema di calcolo interattivo

Sistema di calcolo che permette la comunicazione diretta tra utente e sistema, con tempi di risposta piccoli per l'utente

Sistema di calcolo multiutente

Sistema di calcolo che consente l'utilizzo da parte di più utenti contemporaneamente



Evoluzione dei sistemi operativi

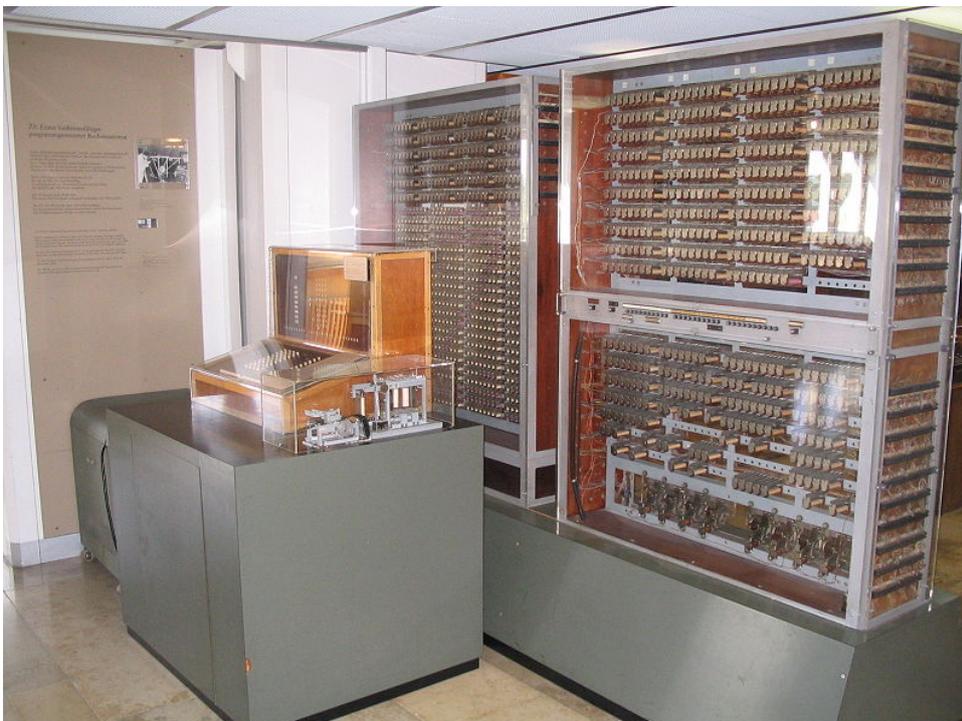
Il **sistema operativo** è un programma di base strettamente correlato all'architettura del **calcolatore elettronico**

Perciò i **sistemi operativi** si evolvono di pari passo con i **calcolatori elettronici**

- I precursori dei calcolatori elettronici:
 - Versuchsmodell-1, -2, -3 e -4
 - Automatic Sequence Controlled Calculator
 - Atanasoff-Berry Calculator
 - Colossus
 - Electronic Numerical Integrator and Computer (ENIAC)
- I primi calcolatori elettronici:
 - Analytical Engine di Babbage
 - Manchester Small-Scale Experimental Machine
 - Electronic Discrete Variable Automatic Computer (EDVAC)
 - Ferranti Mark 1
 - Universal Automatic Computer (UNIVAC)

Versuchsmodell-1, -2, -3 e -4

- Progettati da K. Zuse a Berlino tra il 1936 ed il 1944
- Elettromeccanici, con programma su nastro di celluloide
- Logica binaria, capaci di operazioni in virgola mobile



Automatic Sequence Controlled Calculator

- Progettato da H. H. Aiken tra il 1937 ed il 1942 presso i laboratori IBM
- Donato all'università di Harvard col nome "Harvard Mark I"
- Elettromeccanico, con programma su nastro perforato

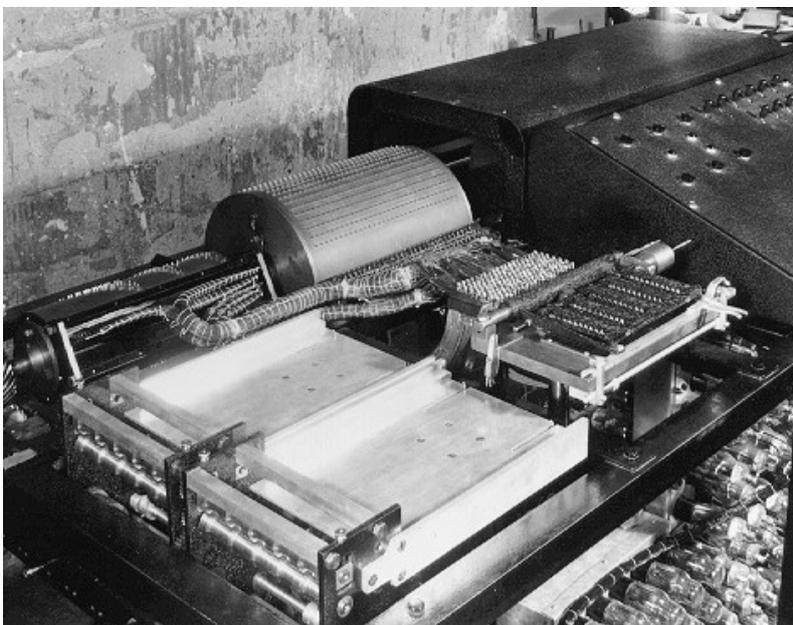


John Kopplin 2002



Atanasoff-Berry computer

- Progettato da J.V. Atanasoff e costruito insieme a C. Berry tra il 1937 ed il 1941 all'università dello Iowa
- Totalmente elettronico e con aritmetica binaria
- Non programmabile e senza salti condizionati

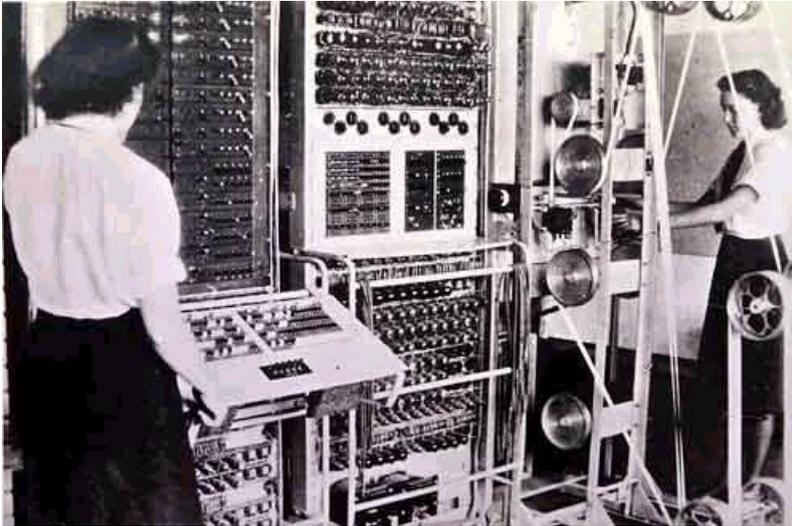


John Kopplin 2002



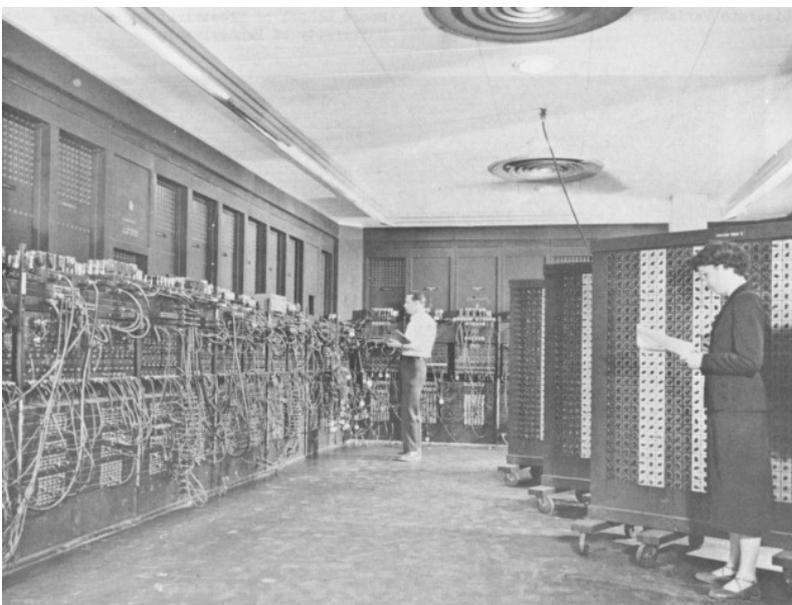
Colossus

- Costruito da T. Flowers nel 1944 con valvole elettroniche
- Impiegato a Bletchley Park (UK) per decifrare i messaggi delle telecriventi e macchine cifranti tedesche



Electronic Numerical Integrator and Computer (ENIAC)

- Ideato da J.W. Mauchly nel 1941
- Realizzato insieme a J.P. Eckert nel 1945 a Filadelfia
- Basato su 18 000 valvole elettroniche



I primi calcolatori elettronici

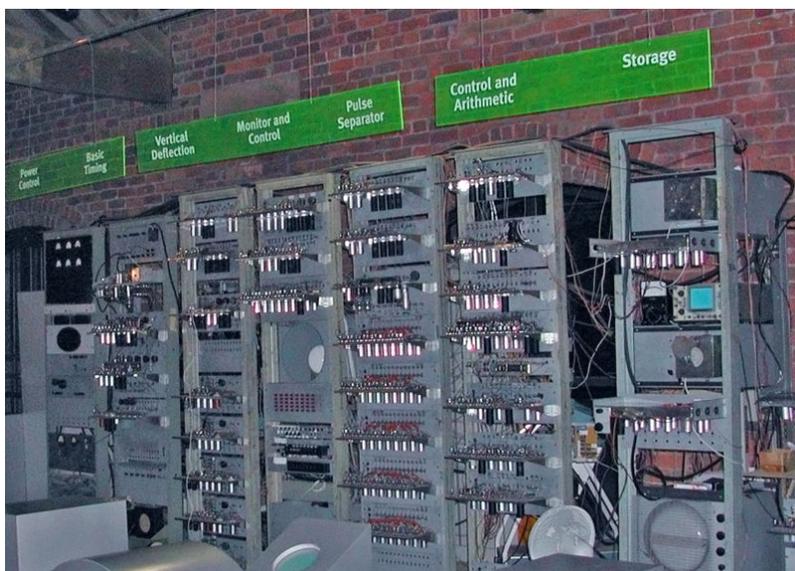
- Prima idea di calcolatore: **Analytical Engine** di C. Babbage (1791–1871)

Il **programma** della macchina è immagazzinato entro una **memoria**, analogamente ai **dati** su cui il programma opera

- Idea ripresa all'inizio del ventesimo secolo da molti scienziati, quali A. Turing, K. Zuse, J.P. Eckert, J. Mauchly, J. von Neumann
- 1948: entra in funzione il primo calcolatore elettronico con programma in memoria centrale: **Manchester Small-Scale Experimental Machine**
- 1949: completamento del progetto **Electronic Discrete Variable Automatic Computer (EDVAC)**
- 1951: primo computer commerciale: **Ferranti Mark 1**, evoluzione del progetto SSEM di Manchester
- 1952: prime consegne dell'**UNIVAC I**, il primo computer commerciale prodotto in USA

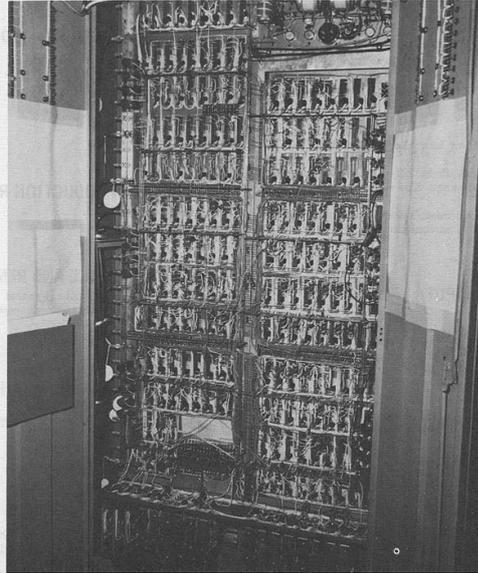
Manchester Small-Scale Experimental Machine (SSEM)

- 1948: primo calcolatore elettronico con programma in memoria centrale ad entrare in funzione
- Costruito da F.C. Williams ed altri all'università di Manchester (UK)
- Memoria di 32 registri a 32 bit, un registro accumulatore ed un registro per l'istruzione corrente



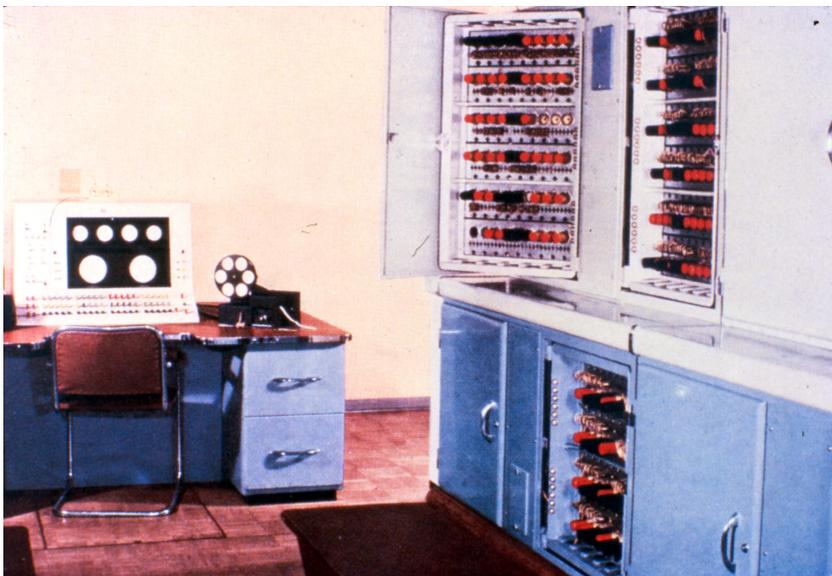
Electronic Discrete Variable Automatic Computer (EDVAC)

- Progetto di Mauchly e Eckert del 1944
- Descritto in un articolo di J. von Neumann del 1945
- Ispiratore del progetto SSEM di Manchester
- Completato solo nel 1949, quando ormai diversi altri calcolatori erano già in funzione



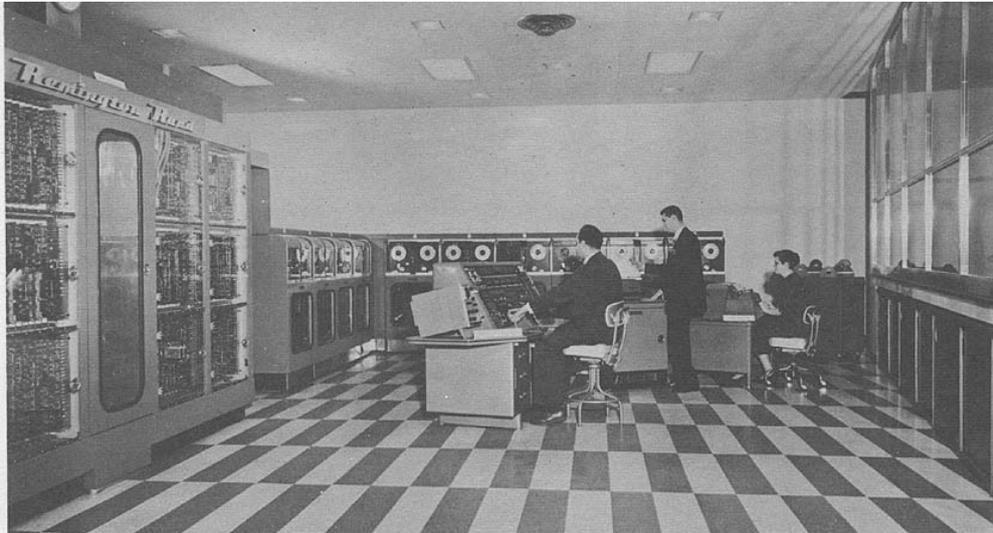
Ferranti Mark 1

- Evoluzione del progetto SSEM di Manchester
- Commercializzato da Ferranti e venduto all'università di Manchester nel febbraio 1951



UNIVAC I

- **UNIVAC I: Universal Automatic Computer**
- Progettato da P. Eckert e J. Mauchly
- 5200 valvole elettroniche, clock a 2.25 MHz, 1905 operazioni/sec
- Costo fino a 1 500 000 \$
- A partire dal 1952, consegnati 46 esemplari
- Non per calcolo scientifico, ma commerciale e statistico



Uso dei calcolatori elettronici di prima generazione

- Macchine estremamente grandi e costose
- L'operatore era spesso anche il programmatore
- Una console di comando consentiva di:
 - Caricare il programma in memoria usando interruttori sulla console (una istruzione alla volta)
 - Caricare il programma in memoria decodificando un nastro di carta od un pacco di schede perforate
 - Impostare l'indirizzo iniziale del programma
 - Avviare l'esecuzione del programma
 - Controllare lo stato d'esecuzione tramite spie luminose
 - Fermare il programma
 - Stampare lo stato dei registri e della memoria
 - Correggere il programma
 - Stampare l'output del programma su nastri di carta, schede perforate, o stampante (telescrivente)



Progressi nell'uso dei calcolatori

Col tempo vennero sviluppati nuovi dispositivi di I/O, quali:

- Nastri magnetici
- Terminali (macchine per scrivere adattate per la console)

Per ciascun dispositivo di I/O si doveva scrivere una procedura speciale per utilizzarlo (**driver di dispositivo**)

Vennero sviluppate **librerie** contenenti i **driver di dispositivi** e altre funzioni comuni che venivano copiate, secondo necessità, nel programma da eseguire

Vennero anche sviluppati nuovi programmi:

- **Assemblatori**, per facilitare la programmazione utilizzando codici simbolici ed etichette
- **Compilatori** per linguaggi quali **FORTRAN** (~1954), **LISP** (~1958), **ALGOL** (~1958), **COBOL** (~1959)

Esempio di procedura per l'esecuzione di un programma

- 1 Installazione del nastro con il compilatore
- 2 Caricamento del programma
- 3 Esecuzione del compilatore
- 4 Rimozione del nastro del compilatore
- 5 Installazione del nastro con l'assemblatore
- 6 Caricamento del file assembler
- 7 Esecuzione dell'assemblatore
- 8 Rimozione del nastro dell'assemblatore
- 9 Caricamento del file oggetto
- 10 Esecuzione del programma oggetto

In caso d'errore si ricominciava da capo!

La maggior parte del tempo il calcolatore elettronico era inattivo, in attesa delle operazioni manuali dell'operatore



Separazione tra operatore e programmatore

- Per diminuire i tempi morti del calcolatore, l'**operatore** divenne una figura professionale differente dal **programmatore**
- I vari **programmatori** consegnavano all'**operatore** un pacco di schede perforate (il **job**, o programma da eseguire)
- L'**operatore** eseguiva uno dopo l'altro tutti i **job** ricevuti
- In caso di errore, l'**operatore** lo segnalava alla fine della giornata al **programmatore**
- Il **programmatore** non poteva più entrare nella stanza del calcolatore, e doveva effettuare il debug esclusivamente sulla base delle informazioni avute dall'**operatore**
- Per diminuire i tempi morti del calcolatore, l'**operatore** suddivideva i **job** in **lotti** (o **batch**)
 - Ad esempio, raggruppava tutti i programmi scritti in FORTRAN in un lotto, e quelli scritti in COBOL in un altro lotto
 - Così caricava una volta sola il nastro con il compilatore FORTRAN ed una volta sola il nastro con il COBOL



Sistemi operativi per i calcolatori di seconda generazione

Negli anni '50 e '60 si diffondono i primi sistemi operativi

- Basati su **monitor residenti**
- Sistemi monoprogrammati
- Schedulazione dei job a lotti
- Con I/O sovrapposto e spooling
- Con semplici **JCL** (Job Control Language)
- Esempi: FMS (Fortran Monitor System), IBSYS (IBM 7094), ATLAS



Monitor residente

I sistemi operativi nascono essenzialmente tentando di automatizzare il lavoro dell'operatore per mezzo di un sequenzializzatore automatico dei lavori di elaborazione:

- Un piccolo programma chiamato **monitor residente** (**batch monitor**) era stabilmente in memoria centrale
- Il **monitor** gestiva il trasferimento automatico dell'elaborazione da un **job** al successivo
- Il **monitor** era informato sulla sequenza dei job da eseguire tramite **schede di controllo** inframmezzate alle schede perforate dei job
- Il **monitor** era perciò costituito da
 - l'**interprete** delle schede di controllo
 - il **caricatore** in memoria dei programmi
 - i **driver** dei dispositivi di I/O (spesso usati anche dai programmi applicativi)

I/O sovrapposto

Il **monitor residente** consentì di ridurre i tempi morti di elaborazione, ma presto il collo di bottiglia divennero i **dispositivi di I/O**

All'inizio degli anni '60 si utilizzavano come dispositivi di I/O

- lettori di schede perforate (per l'input)
- stampanti o perforatori di schede (per l'output)
- nastri magnetici, per velocizzare l'input e l'output
 - le schede perforate venivano riversate su nastro, quando questo era pieno veniva installato sul calcolatore
 - l'output del calcolatore veniva salvato su nastro, quando questo era pieno veniva stampato o riversato su schede
- Era possibile avere diversi sistemi di conversione scheda–nastro e nastro–stampante, così da avere un numero di nastri sempre pronti sufficiente a tenere occupato il calcolatore



Spooling

Il limite maggiore dei nastri è che sono dispositivi ad **accesso sequenziale**

- prima dell'utilizzo debbono essere riavvolti

I **dischi magnetici** non hanno questa limitazione

- hanno perciò soppiantato i nastri magnetici

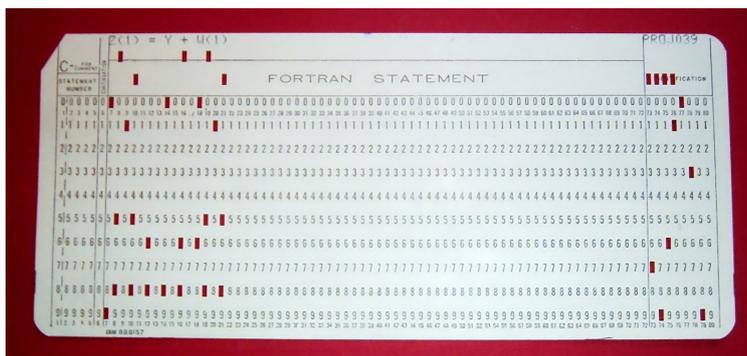
Spooling

La tecnica dello **spooling** (**s**imultaneous **p**eripheral **o**peration **o**n **l**ine) permette di adoperare il **disco magnetico** come una memoria di transito ad alta velocità per i dati di input e di output

Lo **spooling** consente di collegare direttamente al calcolatore i dispositivi di I/O più lenti, perché i trasferimenti vengono effettuati in parallelo al lavoro della CPU

Esempio di job in FMS

- \$JOB** scheda inizio di un job (programma)
- \$FORTRAN** scheda inizio di un programma FORTRAN
- :** schede istruzioni in FORTRAN
- \$LOAD** scheda fine programma, caricamento in memoria
- \$RUN** scheda esecuzione del programma
- :** schede di dati del programma
- \$END** scheda fine dei dati del programma



Sistema operativo ATLAS

- Progettato all'università di Manchester, ~1960
- Costituito principalmente da driver di dispositivi
- Chiamate di sistema realizzate tramite istruzioni speciali
- Sistema operativo a lotti con spooling
- Memoria centrale costituita da un tamburo magnetico da 98K parole da 48 bit
- Memoria cache costituita da memoria a nuclei magnetici da 16K di parole a 48 bit
- Memoria organizzata in pagine da 512 parole
- Meccanismo di memoria virtuale realizzato da memoria associativa di 32 registri
- Trasferimento tra memoria centrale e memoria cache tramite paginazione su richiesta (page on demand)
- Algoritmo predittivo di sostituzione delle pagine in memoria cache basato su un bit impostato ad ogni accesso ad una pagina fisica



Sistemi operativi per calcolatori di terza generazione

Lo sviluppo tecnologico rende i processori sempre più veloci
⇒ aumenta il collo di bottiglia delle periferiche di I/O

Accanto ai mainframe si diffondono **minicalcolatori** come quelli della serie DEC PDP

Tra il 1965 ed il 1980 vengono sviluppati diversi sistemi operativi di nuova generazione:

- Sistemi multiprogrammati (multitasking)
- Partizionamento del tempo di elaborazione (time sharing)
- Offrono servizi di calcolo ad una comunità di utenti
- Interazione tra utente e calcolatore tramite un **terminale**, eventualmente distante e collegato da una linea telefonica
- Esempi: XDS-940, MULTICS, OS/360, UNIX



XSD-940

- Progettato all'università della California di Berkeley, ~1965
- Gestione della memoria con paginazione, solo per rilocazione del codice
- Memoria virtuale di ogni processo di 16K parole
- Memoria fisica di 64K parole, pagine da 2K parole
- Sistema operativo multiprogrammato a partizione di tempo (time sharing)
- Possibilità di condividere tra più processi le pagine fisiche, quando di sola lettura
- I processi erano memorizzati su un tamburo magnetico
- Due modalità d'esecuzione dei processi: modo **utente** (non privilegiato) e modo **kernel** (privilegiato)
- Gestione del tamburo con **file** suddivisi in blocchi da 256 parole
- Chiamate di sistema per creare e distruggere processi
- Comunicazione tra processi tramite condivisione della memoria

MULTICS

- **M**ultiplexed **I**nformation and **C**omputing **S**ervice
- Progettato da MIT, General Electric e Bell Lab., ~1964
- Sistema a partizione di tempo
- Servizio di calcolo fornito ad una città come Boston, analogamente all'elettricità o all'acqua
- Memoria virtuale, con indirizzi costituiti da numero di segmento e offset
- Segmenti suddivisi in pagine da 1K parole da 36 bit
- Ogni segmento era anche un file e vi si poteva accedere tramite il nome del file
- Collegamento dinamico dei segmenti ad un processo
- Il file system era una struttura ad albero a più livelli, estendibile a piacere dagli utenti
- Protezione garantita da
 - una lista di controllo degli accessi associata ad ogni file
 - un insieme di livelli di protezione per i processi
- Supporto a multiprocessore, con riconfigurazione dinamica di CPU, banchi di memoria, dischi, ecc.
- Scritto in linguaggio ad alto livello (PL/1)



OS/360

Il progetto **IBM/360** era una famiglia di calcolatori con macchine di ogni dimensione

Tutti i calcolatori utilizzavano un insieme comune di programmi ed un unico sistema operativo: l'**OS/360**, ~1966

- Sistema operativo estremamente complicato da utilizzare
- Inizialmente scritto in assembler da migliaia di programmatori
- Utilizzava quasi il 50% del tempo di CPU
- Sistema operativo a lotti (il partizionamento di tempo è stato introdotto successivamente)
- Tre varianti principali:
 - **PCP**: mono-programmato
 - **MFT**: multiprogrammato con numero di processi fissato
 - **MVT**: multiprogrammato con numero di processi dinamico

UNIX

- Nel 1969 un gruppo di ricercatori AT&T presso i Bell Lab. (K. Thompson, D. Ritchie, B. Kernighan, e altri) cominciarono a scrivere in assembler un nuovo sistema operativo per un DEC PDP 7
- Alcuni di loro avevano lavorato su MULTICS e decisero di utilizzarne alcune idee:
 - Il file system basato su albero di directory
 - La shell di comandi come processo utente
 - Il sistema di paginazione della memoria virtuale
- Il nuovo sistema fu chiamato **UNICS**, poi cambiato in **UNIX**
- Nel 1970 **UNIX** venne portato su un PDP-11/20
- Nel 1972 **UNIX** venne in larga parte riscritto in linguaggio C, il che lo rese molto facile portarlo su altre architetture di calcolatori
- Poiché AT&T non poteva per problemi di monopolio entrare nel mercato dei calcolatori, **UNIX** fu dato sotto licenza a chiunque lo richiedesse fino al 1983
- La storia di **UNIX** successiva è molto complicata, con centinaia di varianti, citazioni in giudizio, comitati di standardizzazione, ecc.



Sistemi operativi per la quarta generazione di calcolatori

La quarta generazione di calcolatori nasce nei primi anni '70 con i **microcalcolatori**

- I primi erano dei kit che gli appassionati di elettronica compravano e assemblavano da soli
- Molti di essi erano basati sul processore Intel 8080
- Intel commissionò un sistema operativo per l'8080 chiamato **CP/M** (**C**ontrol **P**rogram for **M**icrocomputer)
- Il primo microcomputer di successo fu l'**IBM PC** (**I**BM **P**ersonal **C**omputer) basato sull'Intel 8080 (~1981)
- IBM tentò di utilizzare il **CP/M**, ma non ebbe buona accoglienza, quindi si rivolse ad una piccola azienda, la Microsoft di Bill Gates
- Gates acquistò i diritti di un sistema operativo chiamato **DOS** (**D**isk **O**perating **S**ystem) per 50 000\$
- Gates offrì il **DOS** insieme ad un proprio interprete BASIC all'IBM con il nome di **MS-DOS**

Interfacce utente grafiche

La reale innovazione dei sistemi operativi per la quarta generazione di calcolatori furono le interfacce grafiche

- D. Engelbart inventò la **GUI** (**G**raphical **U**ser **I**nterface) negli anni '60
- L'idea venne incorporata nei prodotti di ricerca della Xerox PARC
- Steve Jobs, che aveva sviluppato il primo computer Apple nel suo garage, vide la **GUI** nei laboratori Xerox
- Il primo tentativo di Jobs fu Lisa: un insuccesso
- Il secondo tentativo di Jobs fu il Macintosh, che ebbe un successo enorme
- L'idea della **GUI** fu ripresa dalla Microsoft, come applicazione sopra il sistema MS-DOS (dal 1985 al 1995)
- Dal 1995 la GUI è divenuta una funzionalità integrata nel sistema operativo Microsoft Windows
- Anche i sistemi operativi di tipo UNIX svilupparono una propria interfaccia grafica chiamata **X Window System**

