Sistemi Operativi

1. Job by Job / Open Shop pre SO era: ≈ fino al 1955
2. Sistemi Batch seconda metà anni ‘50
3. Multiprogrammazione inizio anni ‘60
4. Timesharing metà anni ‘60
5. Sistemi concorrenti fine anni ‘60
6. Sistemi per Personal Computer seconda metà anni ‘70
7. Sistemi Distribuiti anni ’80
8. Sistemi per dispositivi mobili XXI secolo

Fase 1 fino agli anni 50 non solo non si può parlare di SO, ma non si può neanche parlare di programma, in quanto il programma era la configurazione di switch spostati a mano.
L’idea di SO (un programma che ci permette di inserire programmi all’interno del computer) era un concetto inimmaginabile.
Nei primi anni 50 esisteva già l’idea di computer a programma memorizzato (arch Von Neuman).
Quindi il programma veniva scritto (in microcodice eh), poi codifica su schede forate, una volta codificato il programmatore si recava nella sala del computer, le schede venivano organizzate nel corretto ordine ed inserite nel lettore. A questo punto il programmatore poteva sedersi alla console per comandare l’esecuzione del programma.
Tramite opportuni switch era possibile ricevere sulla console precise aree di memoria, a scopo di controllo o di semplice output (na sorta di debugger).
L’output veniva poi stampato su una telescrivente (carta) o su nastro magnetico.
La situazione cambia quando nascono i primi linguaggi Assembler, quindi insieme al programma bisognava caricare il codice dell'assemblatore, in output si otteneva l'eseguibile che poteva a sua volta essere caricato sul computer.
Nasce quindi la necessità di stabilire una figura professionale che si occupasse di gestire tutto ciò.

Fase 2
il compito dell'addetto suggerì una semplice innovazione: permettere al computer di pianificare il lavoro per mezzo di un programma apposito, il Resident Monitor
Nel computer venivano caricati più schede (programmi) in sequenza separate da speciali schede di controllo che si occupavano di definire inizio e fine di un blocco di schede.
Questo Resident Monitor si occupava di controllare l'esecuzione dei programmi. Quando leggeva una scheda di controllo che gli diceva "ESEGUI" eseguiva il programma presente. Quando il programma finiva il controllo tornava al Resident Monitor.
Parlando di schede di controllo, erano schede con un angolo tagliato ed eseguivano funzioni particolari, tipo: $FORTRAN (carica il compilatore fortran e compila il programma), $LOAD (carica il programma in memoria), $RUN (esegui tutte le schede fino alla scheda) $END.
Questo processo venica chiamato "Batch processing".
In particolare cosa si faceva? Si inserivano le schede forate in un lettore che restituiva del nastro magnetico.
Il computer aveva 3 dispositivi: nastro di Input, nastro di sistema e nastro di Output. Il nastro con il programma veniva inserito nel blocco di Input, quando veniva caricato sul nastro di sistema si poteva inserire un'altro nastro in Input. Stessa cosa per l'Output.

Notiamo però una cosa: il Resident Monitor non eseguiva alcun controllo sul programma in esecuzione, ne sui dati e nemmeno sulle eccezioni.
Non vi è il concetto di istruzioni privilegiate o di zone di memoria limitate, sta al programmatore non eseguire programmi "dannosi" come ad esempio accedere a zone di memoria riservate al SO.
In line di massima il programmatore doveva fare attenzione a non: effettuare halt (bloccare il pc), accedere a dispositivi IO, indirizzare locazioni di memoria riservate al sistema.
Ciò è interessante perchè queste sono le regole su cui ancora oggi si basano i sistemi operativa.
Non era possibile implementare queste limitazioni al tempo, perchè richiedevano l'uso di trap, componenti hardware interne ai processori. Gli ingengeri dovevano collaborare con i programmatori.

Fase 3 Multiprogrammazione
nasce dalla necessità di passare oltre alla lentezza dei supporti di memorizzazione, lenti e ad accesso sequenziale; bisognava trovare un modo di sfruttare a pieno i tempi di attesa I/O.
Furono alcune innovazione degli anni '60 a spingere verso la multiprogrammazione. In primis la creazione di dispositivi di memorizzazione di massa (memorie a tamburo e Hard-disk).
In secondo luogo l'invenzione dei transistor che permise processori più potenti con l'aggiunta di un meccanismo per la gestione di interrupt all'interno delle CPU.
Nasce anche il termine Multiprogrammazione, nel 1959 per definire la possibilità di eseguire in modo asincrono (e dunque anche di sovrapporre) diverse operazioni di input e output, NON più programmi contemporaneamente. veniva usato l'hard disk come "buffer", perchè l'hd è ad accesso diretto, mentre i nastri magnetici ad accesso sequenziale. I dati venivano letti dalle schede forate e copiate nell'hardisk, una volta caricato il programma poteva essere lanciato, nel mentre un nuovo pacco di schede veniva letto.
Stessa cosa accadeva per l'output, veniva caricato in HD dalla CPU e poi da li stampato da un altro dispositivo. Questa modalità asincrona di eseguire i programmi e di gestire le operazioni di input e output veniva chiamata spooling.
In pratica un “Multitasking cooperativo”, multitasking ma solo quando un programma lascia la CPU volontariamente (niente scheduler ne interrupt)
Il sistema Atlas, 1961.
Questo era il primo computer ad implementare un SO (al tempo si chiamava supervisore), con due concetti fondamentali di un SO: system call e memoria virtuale (addirittura compresa di Paging).
Atlas aveva un memoria specifica per il codice del supervisor, una seconda memoria per i dati del supervisor ed una core store per contentere il processo utente in esecuzione.
Un’altra memoria a tamburo rotante conteneva codice e dati dei processi attivi ma non in esecuzione, quando un programma iniziava operazioni di I/O veniva spostato su questa memoria.
Il programma poteva essere messo nella memoria tamburo anche in caso richiedesse una pagina non presente nella memoria a tamburo rotante (Paging).
La tecnologia non c’era, ma le idee di base erano già presenti.
B5000 1961
Lo ricordiamo perché il suo supervisore (MPC master control program)implementava tramite particolari registri una memoria di tipo stack per la chiamate delle subroutine. (fu il primo SO a soffrire di thrashing)
Ma anche perché MPC fu scritto in Algol,quindi il primo sistema operativo scritto in un linguaggio di alto livello.

Fase 4, Timesharing
Sistema che interagisce simultaneamente con + utenti. (da differenti console remote)
il primo supervisor timesharing fu il CTSS (compatible tyme sharing system) sviluppato nel 1961 al MIT da un gruppo di ricercatori tra cui Fernando Corbatò (che ha vinto il premio Turing).
Il supervisor alternava l'esecuzione dei vari programmi, concedendo burst di CPU per ciascuno.
Questa idea fu rivoluzionaria per un semplice motivo, il programmatore poteva avviare un programma senza dover fisicamente andare all'elaboratore, ma soprattutto il programmatore poteva interagire con il sistema. MOLTO comodo per il debug (priva veniva solo fatto un core dump, un'immagine della memoria al momento dell'errore)
Le caratteristiche del CTSS avevano molte caratteristiche in comune con in SO moderni:
1 I vari programmi in esecuzione erano tenuti tutti in memoria primaria, ma cio implica la necessità di implementare aree di memoria protette
2 i programmi dovevano poter essere spostati da/verso la memoria secondaria, ergo il codice doveva essere dinamicamente rilocabile (no indirizzi assoluti)
3 veniva implementato un timer hw che permetteva di restituire il controllo al SO finito il quanto di tempo
4 tutte le operazioni di I/O venivano svolte dal SO
5 i programmi che richiedevano operazioni I/O venivano spostati in memoria secondaria
6 gli utenti dovevano poter controllare il loro programma ed il SO dalla loro postazione remota.
All'inizio il CTSS gestiva al massimo fino a 3 utenti, successivamente si arrivò fino a 32 utenti autenticati (quindi con password). Nasce anche il concetto di mail, per comunicare fra diversi utenti
MULTICS
Dal CTSS sempre al MIT nasce il progetto multics, diretto da Robert Fano.
L'obbiettivo MULTICS era proprio creare un SO in grado di fornire a molti utenti l'accesso all'elaboratore.
Il progetto fallisce perché i due finanziatori General Electric (con i suoi Bell Labs) e AT&T abbandonano a causa di una nuova legge sul monopolio che gli impediva di vendere computer.
In ogni caso MULTICS faceva cagare, pesava tantissimo perché era in grado di gestire centinaia di utenti, quindi era lentissimo. Era Overdesigned e overkill per l'hw di quei tempi.
C'è da dire che però il multics gettò le basi dei sistemi operativi moderni, essendo un progetto così grande richiese tantissima ricerca che portò a migliorare la paginazione ed il concetto di file system.
Nel 1965 Robert Daley e Peter Neumann infatti pubblicano un articolo che presenta quello che oggi definiremmo un "file system gerarchico", un sistema indipendente dalla specifica macchina e basato su nomi simbolici (non indirizzi, nomi). La struttura era ad albero gerarchico (nasce il concetto di pathname) e si poteva "uscire" dall'albero tramite opportuni file chiamati link.
Questa idea viene implementata nel 1967 su un sistema chiamato Titan, tramite il primo uso di una nuova tecnica di allocazione: la FAT.
OS/360. Un sistema operativo non time sharing e non innovativo. Ma dominò la scema commerciale, nasce dall'IBM come "middle ground" per rendere compatibile il software e le periferiche di ogni diverso calcolatore IBM. Nasce la linea di prodotti system360, una gamma di prodotti con fasce di prezzo differenti ma che montava un unico OS.
Ma intanto a metà degli anni 80' Unix era già uno standard di riferimento per i sistemi time sharing, lo usavano tutti gli studenti e tutte le università.
Anche perchè lo Unix comparve durante l'espansione dei microcomputer, quindi era pieno di smanettoni che non erano soddisfatti del ready-to-use OS360.

Unix
Dopo l'abbandono del progetto MULTICS da parte di AT&T Ken Thompson e Dennis Ritchie sviluppano autonomamente un OS Multiutente (successivamente diventerà time sharing) con file system gerarchico.
Unix viene riscritto in C nel 1973 (Ritchie è stato uno degli sviluppatori del C) ed i suoi sorgente furono diffusi gratuitamente, ciò ne decretò il suo successo.
Il codice sorgente viene spedito al solo costo del nastro e delle spese di spedizione, secondo la leggenda, con la firma “Love, Ken” (mi piace troppo sta cosa per dimenticarla).
Nel 1977 molte versioni dello Unix erano già state sviluppate o in via di sviluppo, e nel 1978 si contavano più di 600 installazioni.
Nel 1974 Ritchie e Thompson pubblicano sulle Communication of the ACM uno degli articoli più letti e citati di tutta la storia dell’informatica: The Unix Time-Sharing System (è importante perchè è eccezionalmente attuale, le idee li descritte non sono mai profondamente cambiate.)
Unix sarà la base di TUTTI gli OS (a parte quella merda di Windows): linux, OS X, BSD, Solaris.

FASE 5, i sistemi concorrenti.
Negli anni '60 Sistemi Operativi erano troppo complessi, soffrivano di un sacco di problemi, Deadlock, Starvation, thrashing. Durante questa fase i ricercatori cercano proprio di sviluppare software in grado di essere eseguito concorrentemente.
Nascono i semafori, le sezioni critiche e la memoria condivisa.....quindi come non citare Edger Dijkstra, il padre dei semafori. Infatti lui sviluppa il THE nel 1968 (nome in olandese), un sistema costituito su strati sovrapposti di macchine virtuali e sincronizzato tramite semafori.
Degno di nota è un altro sistema sperimentale: RC 4000. il primo progetto di Kernel (da Per Brinch Hansen nel 1969), cioè un nucleo su cio scrivere e personalizzare un prorpio OS.
Hansen è anche il padre del Cuncurrent Pascal, una versione del pascal con sincronizzazione tra processi.

OS per PC
ad inizio degli anni 70 nacquero i primi microcomputer, portandosi dietro un problema: era impossibile fare girare i sistemi operativi che erano stati sviluppati per architetture ben più potenti: il codice del kernel avrebbe occupato più spazio della poca memoria primaria disponibile.
Nel 1969 viene sviluppato l'OS 6 ad Oxford in grado di girare su un pc con 32 kb di memoria principale; implementava un semplice file system, un sistema per la gestione di I/O e niente altro; ci girava un programma alla volta.
Il primo vero OS viene sviluppato da Xerox ed è Alto (come il computer), ed è composto anche da un interfaccia grafica, non legata al sistema operativo però. Questo passo arriva con un progetto dell' 81, lo Xerox Star, che doveva essere l'equivalente elettronico di un ufficio (cartelle, icone, accessori). Purtroppo il computer Star era lento e Xerox non era famosa, quindi vendette poco.

Microsoft
Cioè che ha successo invece ha è la Microsoft, ma bisogna fare un salto indietro:
Nel 1970 Gary Kildall mette a punto un semplice sistema operativo per controllare I floppy: il CP/M, fonda una piccola compagnia, la (Intergalactic) Digital Research. Il CP/M è molto portabile e questo ne determina una enorme diffusione. Ma cosa generava questa sua versatilità? La creazione del BIOS, che permetteva di interfacciare l'hardware con il SO.

Il CP/M era così composto: un interprete di comandi: il CPP, il BDOS cioè una lista di Subroutine che implementava I comandi del CPP (tipo cp, ls ...)ed infine il BIOS che connetteva le funzioni di base dell' hw (dipendendente dalla macchine) alle istruzioni BDOS.
Il 4 aprile del 1975 Paul Allen e Bill Gates fondano la Micro-Soft, scrivendo un compilatore basic per CP/M e poi progettando una scheda di espansione per AppleII.
Quando IBM entra nel mercato PC chiede alla microsoft i diritti per CP/M (pensando che li abbia, dato il progetto della scheda di espansione), Gates li reindirizza verso Kildall (unico possessore dei diritti) ma per qualche motivo l'affare non va in porto.
Gates non si fa scappare l'occasione, compra la licenza d'uso dell'86-dos (variante del CP/M), assume il programmatore che ha creato tale variante (Tim Paterson) e gli chiede di effettuare un "porting" per il processore 8088 che chiama MS-DOS.
Kidall si incazza, fa causa alla IBM ed in realtà vince (nonostante le leggi siano molto poco svilupapte) ma la IBM la risolve vendendo i suoi PC con una variante CP/M, che comunque nessuno compra (perchè costa di più ed è mal supportato)
1985, la microsof è un'azienda già formata che sviluppa Windows: il suo primo applicativo con GUI. Nel 1990 Windows è completamente integrato nel sistema operativo tanto che il 1995 con l'uscita di Windows95 si definisce la fine dei sistemi DOS.

Apple
Apple se la tirava tanto, ma l'AppleII montava Apple-DOS (non MS-DOS), un sistema molto essenziale ispirato a CP/M sviluppato da Steve Wozniak.
La mossa vincente di Apple è quella di rendere pubbliche le caratteristiche software di Apple-DOS, permettendo a programmatori di terze parti di sviluppare software per apple.
Nel 1980 la Apple si lancia nella produzione di sistemi GUI (il primo sarà Lisa, un sistema di merda) con il famoso Machintosh 128k che esce nel 1984.
Sul Mac gira MacOS che sostituiva l'interfaccia a riga di comando ad una a finestre.
Mac Os si sviluppa molto fino alla 10 versione, che viene chiamata MacOS X.
Ma il passaggio da 10 a X non è un semplice rebranding del nome, infatti tutti i sistemi da X in poi diventano Unix-like. Infatti Jobs, esiliato dalla apple nel 1985, torna nel 1997 portando con se un nuovo OS Unix-like

Linux
Partiamo dagli anni 80, nell'83 Richard Stallman al MIT di boston inizia il progetto GNU (GNU is Not Unix): il cui scopo è i dare agli utilizzatori di computer la libertà di usare i loro computer attraverso software che sia stato sviluppato in maniera collaborativa e che tutti siano liberi di usare, condividere, studiare e modificare.
Nel 1985 viene l'intel pubblica il processore 80386 (32 bit) e nel 1987 Tanenbaum crea un SO didattico (MINIX) i cui source sono diffusi con il testo di sistemi operativi ti Tanenbaum.
MINIX è però un sistema a 16 bit, e qui entra in gioco il 22enne Linus Torvalds che volendo un sistema Unix libero per il 386 scrive un nuovo SO usando MINIX come ambiente di sviluppo (ed in realtà basandosi proprio su quest'ultimo), usando come guida un importante libro: The Design of the Unix Operating System, di Maurice Bach.
Il nome in teoria era Freax, ma un amico di Linus quando carica i file sul server FTP dell'università lo chiama Linux. Perchè è importante GNU? Perchè Linux è protetto da licenza GNU GPL (General Public License) e questa libertà di modifica ed utilizzo fu il successo di Linux.

Al momento Linux è gestito dalla Linux Foundation, associazione no-profit ed ha migliaia di sottoprogetti.

Fase 7: sistemi distribuiti
Un OS che unisce più computer collegati fra di loro da una rete in modo tale che ogni utente collegato ad uno di questi nodi abbia l'illusione di usare una sola unità computazionale.
Tutto è distribuito: dai dati (filesystem) all' elaborazione. L'utente apre un file o lancia un programma e il SO decide come bilanciare il carico di lavoro andando a leggere /operare sul nodo che meglio crede.
Un vantaggio è che ha un'enorme tolleranza ai guasti, se un nodo si guasta non c'è problema.
Questi sistemi si diffondono negli anni 80 usando un meccanismo chiamato RPC (Remote Procedure Call)
la prima implementazione di un sistema distribuito si ha nel 1981 grazie a Xerox mentre nel 1984 nasce NFS (Network File System), un file system distribuito. In ogni caso i sistemi distribuiti sono sempre rimasti relegati alla ricerca ed alla sperimentazione (due sistemi importanti sperimentali sono lo Unix United System del 1982 ed Amoeba)

Fase 8: sistemi per dispositivi mobili
Nella seconda metà degli anni '90 iniziano ad essere costruiti i primi dispositivi mobili, quindi nasce la necessità di creare un SO che li possa far funzionare, con particolare attenzione alla scarsa memoria ed alla scarsa batteria.
La prima azienda a buttarsi in questo campo è la Nokia con il Nokia 9000 ed il MessagePad (era un palmare).
Dal 2000 in poi nascono un sacco di sistemi operativi: Android, Apple iOS, Windows Mobile, Symbian e Blackberry.
In particolare Android nasce nel 2003 basto su un kernel Linux ma sviluppato in Java e nel 2005 viene acquistato da Google, è un sistema Open Source (davvero??) ed è il sistema più usato (80% dei dispositivi mobili)
Invece Apple iOS viene lanciato nel 2007 ed usa un kernel XNU (Xnu is Not Unix) derivato dal sistema NextSTEP progettato da Jobs nel 1996.