

Sistemi Operativi

Ionut Zbirciog

5 October 2023

1 Cos'è un Sistema Operativo?

Un software che si interfaccia con i componenti hardware della macchina (CPU, MEMORIA, MEMORIE NON VOLATILI, DISPOSITIVI I/O).

2 Componenti di un calcolatore moderno

- Uno o più processori
- Memoria principale
- Dischi e unità flash
- Periferiche I/O

2.1 Doppia modalità supportata dall'hardware

Un'applicazione è un processo che esegue delle istruzioni. Il sistema mette a disposizione delle funzioni (chiamate di sistema). Il sistema operativo si prende carico di gestire le risorse ed eseguire le istruzioni.

- Modalità kernel (supervisor)
- Modalità utente

3 Il SO come una macchina

- Idea di Astrazione: SO si pone tra l'hardware e le applicazioni (l'obiettivo del SO è di astrare l'hardware).
- Visione top-down: SO fornisce astrazioni ai programmi applicativi.
- Vista Bottom-up: SO gestisce parti di un sistema complesso, fornisce un'allocazione ordinata e controllata delle risorse.

4 Il SO come gestore di risorse

Il SO permette di gestire le risorse per:

- Eseguire più programmi in esecuzione
- Supportare più utenti

Il multiplexing permette di mettere a disposizione delle risorse in modo condiviso, sia nel tempo (CPU, Stampante) che nello spazio (Memoria centrale, Disco).

5 Breve Storia dei Sistemi Operativi

1. First Generation: Vacuum Tubes

- Tubi vuoti che emettevano luce per simulare gli 1 e gli 0.
- Nessun supporto per la programmazione; l'unico modo era spostare i connettori.
- I calcoli potevano essere eseguiti uno alla volta (limitata ottimizzazione della macchina).

2. Second Generation: Transistors and batch systems

- Scomposizione delle operazioni della macchina e del sistema operativo in blocchi operativi.
- Spiegazione delle operazioni batch.
- Componenti come il lettore di schede (per programmi, compilatori e il sistema operativo) e il lettore di nastri (da schede a nastri).
- I programmi e i dati da elaborare venivano trascritti e inviati all'esecuzione.

3. Third Generation: Integrated Circuits and Multiprogramming

- Caricamento di applicazioni contemporaneamente grazie a memorie più grandi.
- Partizionamento della memoria.
- Spooling: caricamento dei lavori senza interruzioni.
- Time Sharing: la CPU viene assegnata a lavori o utenti diversi mentre è in attesa.
- Non è stata implementata fino agli anni '70 a causa della mancanza di protezione hardware per garantire che, in caso di errore, un processo non scrivesse in un'area riservata ad un altro.

4. Fourth Generation: Personal Computer

5. Mobile Computers: Smartphone

6 UNIX

SO multiutente e multiprogrammazione, open-source. Il linguaggio C fu sviluppato per scrivere UNIX. Standard POSIX sviluppato dall'IEEE per garantire compatibilità inter-sistemi. Da UNIX derivano molti altri sistemi operativi.

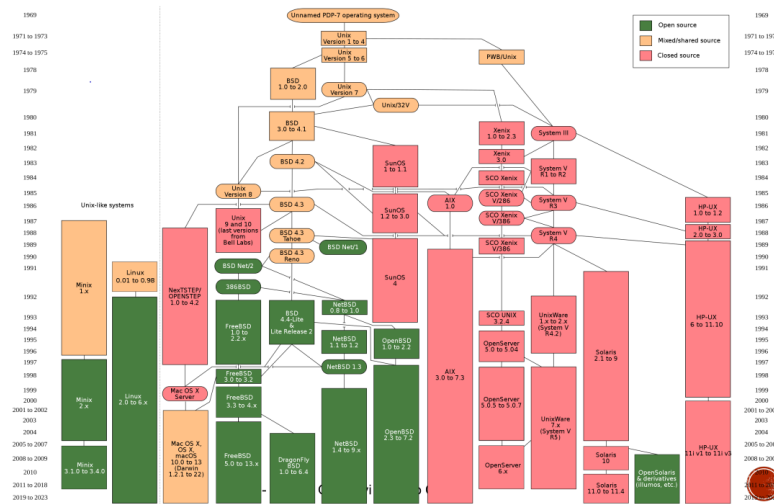


Figure 1: Sistemi operativi derivati di UNIX

6.1 MINIX - una variante di UNIX

Un piccolo sistema scritto da Tanenbaum, compatibile con gli standard UNIX.

6.2 Da MINIX a LINUX

Hello everybody out there using minix -

I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) A T clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).

I've currently ported bash(1.08) and g c c(1.40), and things seem to work. This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :-)

Linus (torvalds@kruuna.helsinki.fi)

P.S. Yes - it's free of any minix code, and it has a multi-threaded fs. It is NOT portable (uses 386 task switching etc), and it probably never will support anything other than A T-harddisks, as that's all I have :-)

Figure 2: Il messaggio di Linus Torvalds

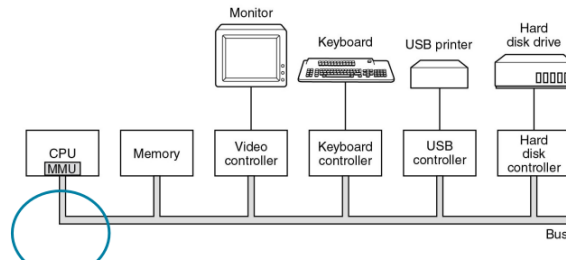


Figure 3: Architettura di un calcolatore

7 Uno sguardo all'hardware

7.1 Il processore

Il ciclo della CPU: preleva (fetch), decodifica (decode), esegue (execute).

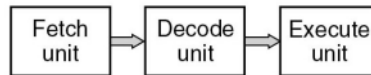


Figure 4: fetch - decode - execute

- Registri
 - Program Counter: indica l'istruzione successiva
 - IR: indica l'istruzione che viene eseguita
 - Stack Pointer: punta alla cima dello stack della memoria
 - Program Status Word (PSW): contiene informazioni sullo stato del programma, fondamentale per chiamate di sistema e I/O.
- Multiplexing: il sistema operativo esegue programmi in modo efficiente
- Pipeline: Esegue in parallelo istruzioni che possono essere eseguite a livello circuitale. Anche in caso di un'istruzione condizionale, la pipeline esegue anche l'operazione successiva pur di non fermarsi dall'eseguire istruzioni.
- Più di un processore: Più processori fisici o logici, multithreading.

7.2 La memoria di un calcolatore

Problemi del sistema cache:

- Quando inserire un nuovo elemento nella cache?
- In quale riga della cache inserire il nuovo elemento?

- Quale elemento rimuovere dalla cache quando è necessario uno slot.
- Deve mettere un elemento appena eliminato nella memoria più grande.

Nota: Più la memoria è veloce, più è piccola

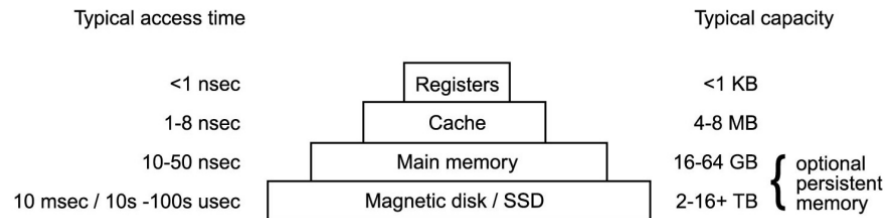


Figure 5: Gerarchia della memoria

7.3 Dispositivi di I/O

- Il controller: più semplice da usare per il SO, ogni controller ha bisogno di un driver.
- Il dispositivo: interfaccia elementare ma complicata da pilotare

Driver: pezzetto di sistema operativo che non viene fornito con il sistema operativo, deve essere inserito all'interno del sistema operativo e avere tutti i diritti del sistema operativo. L'unico momento in cui il sistema operativo concede al driver tutti i diritti è al momento dell'avvio, infatti dopo aver installato un driver, il sistema operativo chiede all'utente di riavviare il sistema. In Linux hanno dovuto forzare questa cosa a causa delle porte USB.

Il driver interagisce con il controller per:

- Eseguire l'I/O
 - il processo esegue la chiamata di sistema
 - il kernel effettua una chiamata al driver
 - il driver avvia l'I/O
- Interrogare il dispositivo per vedere se ha finito oppure chiede al dispositivo di generare un interrupt quando ha finito

7.4 Il DMA

DMA (Direct Memory Access) consente ai componenti di accedere direttamente alla memoria del computer senza coinvolgere la CPU, migliora l'efficienza ed aumenta le prestazioni nelle operazioni di I/O.

7.5 Buses

- Dispositivi legacy collegati a un processore hub separato
- USB è stato sviluppato per connettere dispositivi lenti al computer
- La USB deve sia alimentare che comunicare

7.6 Avvio del sistema

Quando si accende la macchina, si legge la ROM con le istruzioni per tutte le periferiche. Il BIOS esegue i comandi e dà retta al BOOTLOADER che legge il sistema operativo in base alla periferica.