

Introduzione ai sistemi operativi

Pietro Braione

Reti e Sistemi Operativi – Anno accademico 2021-2022

Cos'è un sistema operativo?

È il primo programma che viene eseguito quando viene acceso il computer

Ci permette di eseguire tanti programmi contemporaneamente, e di far scambiare informazioni tra di loro

Ci permette di gestire il computer, ed in particolare di installare e mandare in esecuzione i programmi veramente utili (applicazioni)

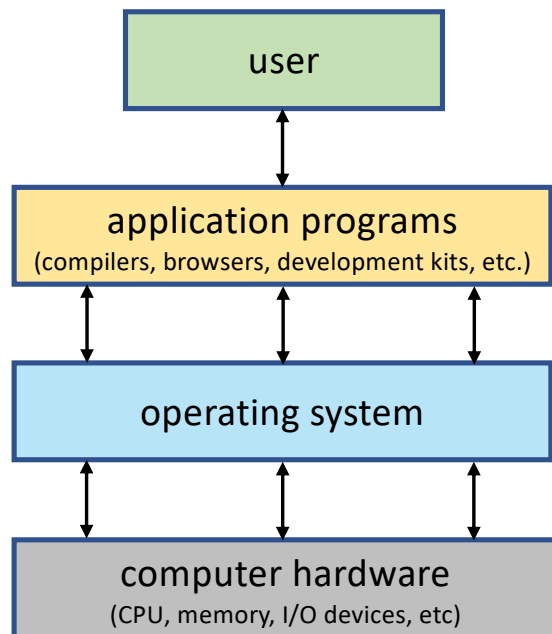
Fornisce un ambiente omogeneo, ed un insieme di «regole» (ad esempio, regole grafiche) alle quali chi sviluppa applicazioni deve attenersi perché le applicazioni si integrino in questo ambiente

Mantiene ed organizza i nostri dati sotto forma di file e cartelle

Cos'è un sistema operativo?

- È un insieme di programmi (software)
- Che gestisce gli elementi fisici di un computer (hardware)
- Fornendo una piattaforma di sviluppo ai programmi applicativi che permette loro di **condividere** ed **astrarre** le risorse hardware
- E agendo da intermediario tra utenti e computer fisico, permettendo agli utenti di **controllare** l'esecuzione dei programmi applicativi

I quattro componenti di un sistema di elaborazione



- Utenti: persone, macchine, altri computer...
- Programmi applicativi: risolvono i problemi di calcolo degli utenti
- Sistema operativo: coordina e controlla l'uso delle risorse hardware
- Hardware: risorse di calcolo (CPU, periferiche, memorie di massa...)

Di cosa si occupa un sistema operativo?

- Di due aspetti molto diversi, apparentemente scorrelati:
- Da un lato, **astrae** le risorse hardware del computer, presentando agli sviluppatori dei programmi applicativi una «macchina estesa» più facile da programmare (ad esempio, files al posto di blocchi del disco)
- Dall'altro, **gestisce e multiplexa** le risorse hardware del computer, assegnandole ai programmi in maniera equa ed efficiente e controllando che questi le usino correttamente
- Quest'ultimo aspetto nasce dalla necessità di sfruttare efficientemente il computer eseguendo più programmi per più utenti

Origine dei sistemi operativi (1)

- I computer della prima generazione (1945-1955) erano programmati manualmente (schede e spinotti), ed erano più spesso guasti che funzionanti: spesso non vi era il software, né vi era l'esigenza di sfruttarli efficientemente (le persone usavano i computer **interattivamente**)
- Con l'avvento dei transistor, i computer della seconda generazione (1955-1965) diventano più affidabili, e nascono i linguaggi di programmazione ad alto livello e i loro compilatori (FORTRAN, 1954)
- Inizia a diventare importante ridurre il più possibile i tempi morti per ottimizzare l'uso dei costosi computer

Origine dei sistemi operativi (2)

- Idea: sostituire l'uso interattivo con l'elaborazione **batch** (a lotti):
 - Un job, nella forma di un pacco di schede perforate, viene letto da un computer a basso costo, e trasferito su nastro
 - Si prosegue così per una certa insieme di job fino a saturare il nastro
 - Il nastro viene caricato sull'elaboratore, che esegue tutti i job e produce l'output su un altro nastro
 - Un terzo computer a basso costo, collegato ad una stampante, legge il nastro di output e stampa i risultati di tutti i job
- Nascono i programmi monitor per il caricamento ed esecuzione dei job:
 - Ogni job ha delle schede di controllo («esegui il compilatore FORTRAN», «esegui il programma appena compilato»...)
 - L'elaboratore esegue il programma monitor per interpretare il successivo comando nel job control language, più il programma corrente da eseguire, e così via
- Questi programmi monitor sono gli antesignani dei sistemi operativi

Origine dei sistemi operativi (3)

- I computer della terza generazione (1965-1971) erano basati sui primi circuiti integrati
- IBM con la linea 360 (1965) cerca di creare un'unica linea di computer adatta sia per i calcoli scientifici sia per le ditte commerciali
- I computer della linea 360 erano **compatibili** tra di loro e differivano solo per prezzo e prestazioni
- Questo introduceva alcuni problemi:
 - Lo stesso software doveva funzionare su tutti i computer della linea, e i software sviluppati dagli utenti dovevano essere portabili da un computer all'altro
 - Occorreva gestire le risorse della macchina in maniera efficiente in tutti gli scenari applicativi
- Si pone il problema di fornire un ambiente identico per tutti i computer: il sistema operativo OS/360 diventa il primo sistema operativo per una *famiglia* di computer

Origine dei sistemi operativi (4)

- L'input/output diventa molto più lento dell'elaborazione
- Viene sviluppata la tecnica della **multiprogrammazione**:
 - Molti programmi applicativi caricati in memoria contemporaneamente
 - Quando un programma è impegnato nell'I/O il processore passa ad eseguirne un altro
- Viene anche sviluppata la tecnica dello **spooling**:
 - L'I/O viene mediato attraverso un buffer in memoria centrale
 - Il programma interagisce con il buffer (più rapido) anziché con la periferica
 - Esempio: code di stampa
- Necessità di nuovi meccanismi hardware per supportare queste tecniche:
 - Protezione della memoria
 - Interrupt

Origine dei sistemi operativi (5)

- La modalità di lavoro batch aumenta l'efficienza di uso del computer, ma è scomoda e inefficiente per le persone rispetto a quella interattiva
(<https://www.computerhistory.org/revolution/punched-cards/2/211/2253>)
- Ma se lasciar utilizzare il computer interattivamente a una sola persona ne comporta un sottoutilizzo, permetterlo a più persone può risolvere il problema:
 - Un singolo utente alterna lunghi periodi di inattività a momenti di intensa attività
 - Se ho molti utenti interattivi, i periodi di inattività di ciascuno si compensano con quelli di attività degli altri, e il computer risulta utilizzato
- Questo porta allo sviluppo della tecnica del **time-sharing**:
 - Una versione avanzata della multiprogrammazione
 - Gli utenti hanno a disposizione dei terminali in linea
 - Il sistema operativo alterna l'esecuzione dei diversi programmi attivi in maniera che ogni utente ha l'impressione di operare in maniera interattiva
- Nascono addirittura compagnie che offrono servizi di time-sharing computing attraverso modem
- Sistemi operativi importanti: CTSS (il primo time-sharing), Multics, Unix

Origine dei sistemi operativi (6)

- La quarta generazione di computer (1971-oggi) è basata sui circuiti integrati a larga scala (VLSI)
- I computer diventano così poco costosi che nasce il concetto di **personal computing**: ogni persona può avere un computer dedicato (personal computer, o PC)
- I primi sistemi operativi per i personal computer non implementano la multiprogrammazione o il time-sharing
 - Tali computer avevano troppe poche risorse di calcolo e processori troppo semplici per implementarli
 - Inoltre a che scopo, se il computer deve essere usato da una persona sola?
- Diventa invece importante semplificare l'interazione per gli utenti meno esperti: nascono le **interfacce utenti grafiche (GUI)**
- Negli anni più recenti è aumentata la potenza di calcolo e si è sviluppata l'Internet globale e il mobile computing
 - I personal computer diventano sempre più potenti e i loro sistemi operativi sempre più simili ai sistemi time-sharing come Unix
 - I dispositivi mobili hanno un'altra risorsa scarsa da economizzare: la durata della batteria
- Sistemi operativi importanti: Xerox Alto e Star, DOS, Windows NT, macOS, Linux, Android, iOS

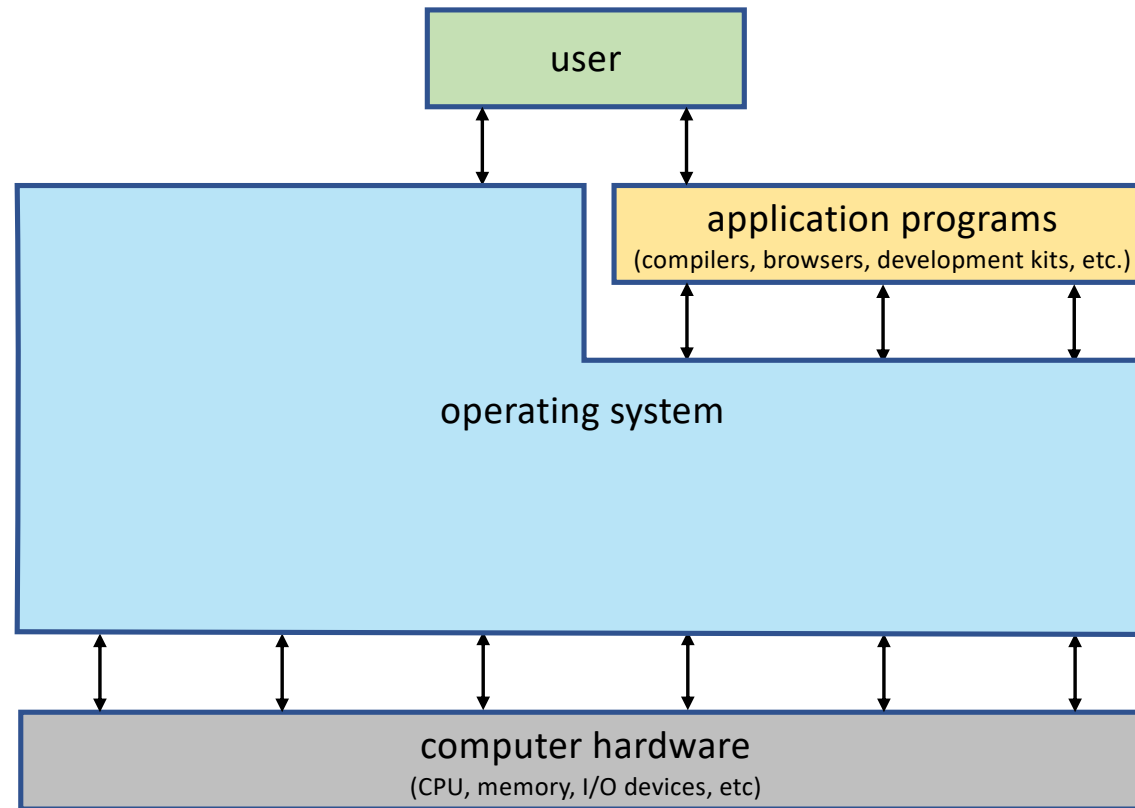
Che caratteristiche deve avere un sistema operativo?

- Server, mainframe: massimizzare la performance, rendere equa la condivisione delle risorse tra molti utenti
- Laptop, PC, tablet: massimizzare la facilità d'uso e la produttività della singola persona che lo usa
- Dispositivi mobili: ottimizzare i consumi energetici e la connettività
- Sistemi embedded: funzionare senza (o con minimo) intervento umano e in tempo reale
- La **maledizione della generalità**: se il sistema operativo deve supportare una classe di applicazioni troppo ampia, finisce per non essere in grado di supportare nessuno dei suoi molteplici scenari d'uso particolarmente bene

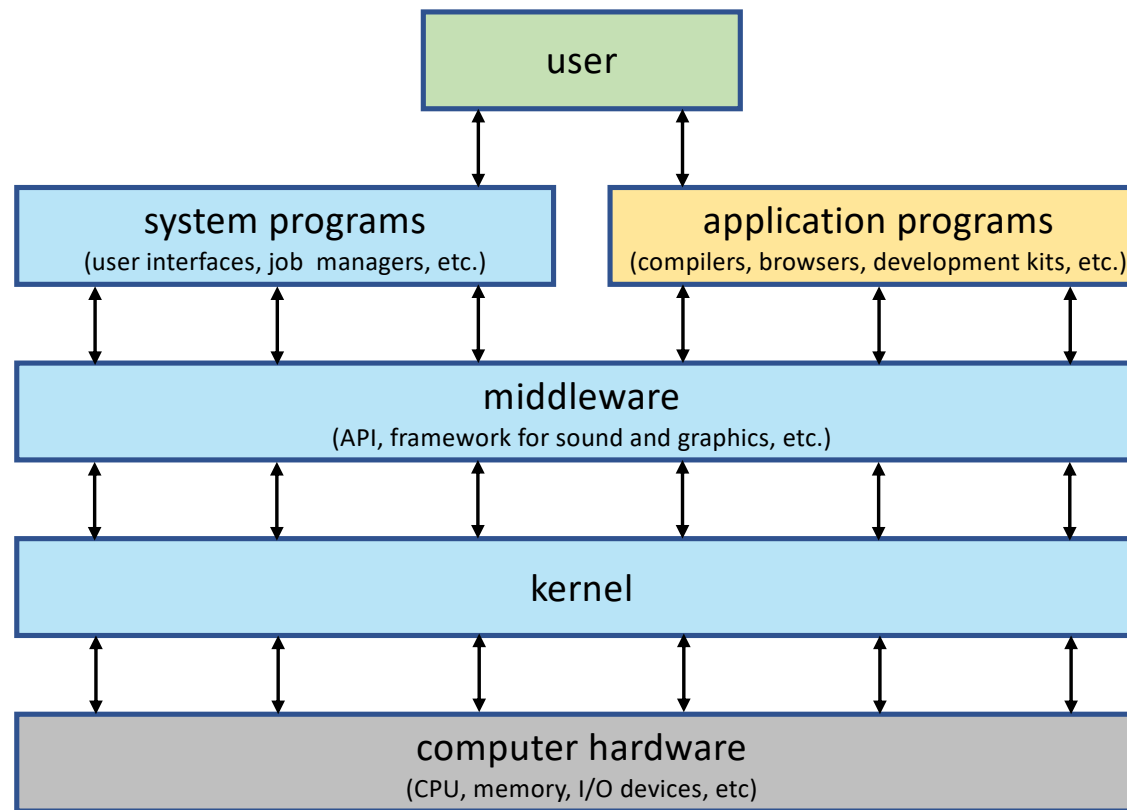
Da che programmi è composto un sistema operativo?

- Non c'è una definizione universalmente accettata
- In generale però un sistema operativo comprende almeno:
 - **Kernel:** il «programma sempre presente», che fornisce i principali servizi di gestione dell'hardware (processore, memoria, periferiche, dischi...)
 - **Programmi di sistema:** non sempre in esecuzione, offrono ulteriori funzionalità di supporto (gestione di jobs e processi, interfaccia utente...)
 - **Middleware:** servizi di alto livello per la programmazione di applicazioni (API, framework per grafica, per suono...)

I quattro componenti di un sistema di elaborazione, rivisitati



I quattro componenti di un sistema di elaborazione, rivisitati



Gestione delle risorse

- Il sistema operativo deve gestire diversi tipi di risorse, sia hardware che astratte (ossia realizzate dal sistema operativo stesso):
 - Processi
 - Memoria
 - Files
 - Memorie di massa
 - Cache
 - I/O

Gestione dei processi (1)

- Un **processo** è un *programma in esecuzione* nel sistema
 - È l'unità di lavoro del sistema
 - Un processo è un'entità *attiva*, mentre un programma è un'entità *passiva*
- I processi hanno bisogno a loro volta di diverse risorse per poter completare le proprie operazioni (CPU, memoria, I/O, files...)
- Possono avere uno (single-threaded) o più di un (multi-threaded) flusso di esecuzione del programma concorrenti

Gestione dei processi (2)

- Un sistema di elaborazione ha di solito molti più processi che CPU: i processi operano in maniera concorrente condividendo le CPU disponibili
 - Processi utente
 - Processi del sistema operativo
- Attività di gestione processi del sistema operativo:
 - Creazione e cancellazione dei processi utente e di sistema
 - Schedulazione dei processi e thread sulle CPU
 - Sospensione e ripristino dei processi
 - Meccanismi di sincronizzazione tra processi
 - Meccanismi di comunicazione tra processi
 - Meccanismi di individuazione e risoluzione dei deadlock

Gestione della memoria

- Per eseguire un programma le sue istruzioni e dati (o parte di essi) devono essere nella memoria centrale
- Il sistema operativo stabilisce cosa deve essere in memoria e quando
- Attività di gestione della memoria:
 - Tenere traccia di quali parti della memoria sono usate e da chi
 - Decidere quali processi (o parti di essi) caricare in memoria e quali trasferire nelle memorie di massa
 - Assegnare e revocare memoria ai processi secondo necessità

Gestione dei file

- I sistemi operativi astraggono le caratteristiche dei dispositivi di memorizzazione
 - Gestiscono dispositivi diversi con caratteristiche diverse (tempo d'accesso, capacità, modalità d'accesso...)
 - Presentano un'interfaccia logica uniforme basata sui **file**, che sono raccolte di informazioni correlate
 - Organizzano i file in directory
- Attività di gestione dei file:
 - Creazione e cancellazione dei file e delle directory
 - Modifica e manipolazione di file e directory
 - Associazione file a dispositivi di memoria secondaria
 - Creazione di copie di riserva (backup)

Gestione delle memorie di massa

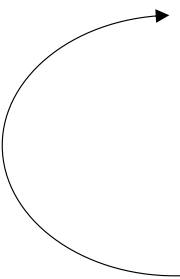
- Le memorie di massa sono usate:
 - Per memorizzare programmi e dati in maniera permanente
 - Per memorizzare programmi e dati che la memoria centrale non riesce a contenere
- Le performance del sistema dipendono in maniera fondamentale dalla performance del sottosistema di gestione delle memorie secondarie
- Attività di gestione delle memorie di massa:
 - Montare e smontare le unità di memoria
 - Gestione dello spazio libero
 - Assegnazione dello spazio
 - Scheduling del disco (minimizzazione del tempo di lettura)
 - Partizionamento
 - Protezione

Gestione delle cache

- Il **caching** è un concetto importante ed implementato a diversi livelli (nell'hardware, nell'OS, nelle applicazioni)
- L'informazione in uso (o che si ritiene verrà usata a breve) viene temporaneamente copiata dalla memoria più lenta a quella più veloce
- Per accedere all'informazione dapprima viene consultata la memoria più veloce (**cache**) per vedere se ne contiene una copia
 - Se la contiene, viene usata (accesso rapido)
 - Altrimenti, si consulta la memoria più lenta (accesso lento)
- Dal momento che le cache sono molto più piccole della memoria che replicano, le politiche di sostituzione del contenuto delle cache influiscono in maniera determinante sulla performance

Caratteristiche dei diversi tipi di memoria

Level	1	2	3	4	5
Name	registers	cache	main memory	solid-state disk	magnetic disk
Typical size	< 1 KB	< 16MB	< 64GB	< 1 TB	< 10 TB
Implementation technology	custom memory with multiple ports CMOS	on-chip or off-chip CMOS SRAM	CMOS SRAM	flash memory	magnetic disk
Access time (ns)	0.25-0.5	0.5-25	80-250	25,000-50,000	5,000,000
Bandwidth (MB/sec)	20,000-100,000	5,000-10,000	1,000-5,000	500	20-150
Managed by	compiler	hardware	operating system	operating system	operating system
Backed by	cache	main memory	disk	disk	disk or tape



Lo spostamento dei dati tra i diversi livelli della gerarchia di memoria può essere implicito (automatico) o esplicito

Caching in presenza di concorrenza

- In presenza di concorrenza, ogni entità concorrente può avere una cache distinta
- È necessario garantire la **coerenza delle cache**, ossia far sì che attraverso le rispettive cache le diverse entità concorrenti abbiano una visione coerente della memoria sottostante
- I sistemi multiprocessori possono avere più cache separate: Ad esempio, due core distinti hanno tipicamente due cache di livello 1 distinte. L'hardware garantisce la coerenza in maniera che tutte le CPU abbiano nella propria cache il valore più recente
- Nei sistemi multitasking occorre stare attenti che ogni processo usi sempre il valore aggiornato più di recente. Questo deve essere garantito dal sistema operativo
- Nei sistemi distribuiti (ad esempio, i cluster) mantenere la coerenza con copie multiple di uno stesso dato può essere molto difficile

Gestione dell'I/O

- Uno dei compiti dei sistemi operativi è nascondere all'utente le peculiarità dei diversi dispositivi
- Il **driver** è la parte di software dell'OS che conosce le specificità di un dispositivo; i driver presentano tutti una stessa interfaccia omogenea
- Attività di gestione dell'I/O:
 - Gestione della memoria dell'I/O:
 - Buffering: memorizzazione dei dati mentre sono trasferiti
 - Caching: copia dei dati in unità di memoria ad accesso più rapido
 - Spooling: sovrapposizione e disaccoppiamento delle operazioni di I/O
 - Driver dei dispositivi specifici (possono essere moltissimi!)
 - Interfaccia omogenea e generale per i driver

Altri due argomenti importanti

- Protezione e sicurezza
- Virtualizzazione

Protezione e sicurezza

- **Protezione:** meccanismi di controllo di accesso alle risorse del computer
- **Sicurezza:** difesa da attacchi esterni ed interni al sistema
- La principale funzionalità che i sistemi offrono è la capacità di distinguere tra gli utenti e di assegnare permessi agli utenti
 - Le identità utente (**user IDs**) includono nome e un numero identificativo
 - L'user ID è associata ai file e i processi di quell'utente per determinare il controllo di accesso
 - Le identità di gruppo (**group IDs**) permettono di definire insiemi di utenti e di associare loro opportuni controlli d'accesso
 - In certi contesti può essere necessario permettere agli utenti di **scalare i privilegi**, ossia ottenere temporaneamente permessi aggiuntivi per certe attività

Virtualizzazione

- Permette di eseguire applicazioni per sistemi operativi differenti
- Idea: separare multiprogrammazione da astrazione:
 - Creare molteplici macchine virtuali concorrenti su uno stesso hardware (multiplexing)...
 - ...ma identiche nelle funzionalità all'hardware su cui eseguono (nessuna astrazione)
- Un **virtual machine monitor** (VMM, o **hypervisor**) fornisce servizi di virtualizzazione realizzando le macchine virtuali:
 - VMM tipo 0: nel firmware
 - VMM tipo 1: direttamente in esecuzione sull'hardware
 - VMM tipo 2: installato su un sistema operativo (host)
- Ogni macchina virtuale può eseguire un diverso sistema operativo guest