

Introduzione generale

Cos'è un sistema operativo

Il **sistema operativo** (*operating system*) è un insieme di programmi che interagiscono e controllano le funzionalità della macchina fisica (*hardware*) in modo tale da offrire ai programmi applicativi ed agli utenti un insieme di funzionalità con un elevato livello di astrazione. I programmi applicativi accedono quindi alle risorse *hardware* mediante opportuni componenti del sistema operativo. Inoltre il sistema operativo offre un'interfaccia amichevole dalla quale l'utente può invocare i principali servizi.



Il **sistema operativo** aggiunge un livello (*software*) alla macchina fisica realizzando così una nuova macchina virtuale che facilita l'uso delle risorse fisiche, consente di superare alcuni problemi dovuti alla loro limitazione (ad esempio dimensioni della memoria centrale) e ne ottimizza e regola l'accesso (ad esempio evitando conflitti).

Il concetto di macchina virtuale

Una macchina virtuale è un dispositivo di calcolo astratto caratterizzato da:

1. un linguaggio che la macchina è in grado di capire, specificato da una opportuna sintassi, ovvero da un insieme di regole che definiscono i costrutti corretti del linguaggio;
2. un interprete per tale linguaggio, ovvero un meccanismo che permetta di eseguire le istruzioni specificate nel linguaggio definendone così anche il significato, ovvero la semantica.

Il concetto di macchina virtuale è ampiamente usato in molti settori dell'informatica sia per descrivere sistemi che per specificare le caratteristiche dei linguaggi di programmazione.

Ad esempio possono essere considerate macchine virtuali:

1. una macchina fisica, ovvero un calcolatore *hardware*, definito da un opportuno linguaggio di programmazione (linguaggio macchina) e dall'interprete realizzato dal ciclo *fetch-execute* della CPU;
2. un editor di testi, definito dai comandi che agiscono sull'astrazione di un documento elettronico.

Le macchine astratte possono essere combinate in vario modo per costruire macchine più complesse.

Architettura di un sistema operativo

I componenti di un sistema operativo si possono considerare organizzati a livelli secondo la struttura illustrata qui sotto. Tale struttura definisce una gerarchia di macchine virtuali: i componenti di uno

stesso livello realizzano una macchina virtuale che, usando i servizi del livello sottostante, definisce nuove funzionalità.



Componenti di un sistema Operativo 1

Interprete dei Comandi (o *shell*)

Fornisce l'interfaccia con l'utente che attraverso di esso può inviare comandi al sistema operativo (ad esempio lettura dalla memoria di massa, creazione ed attivazione di un processo, allocazione di memoria centrale, caricamento di un programma in memoria centrale ecc.). L'interfaccia dell'interprete con l'utente può essere di tipo testuale oppure grafico.

Gestore dei *file* (*file system*)

Componente che permette l'organizzazione delle informazioni in strutture logiche, dette **file**, identificati da nomi e accessibili all'utente tramite opportuni comandi (apertura, lettura, scrittura, chiusura). Gestisce le operazioni di allocazione di memoria di massa necessarie per memorizzare i *file*

Gestore delle periferiche

Componente che fornisce una visione astratta delle periferiche e dei relativi comandi di input/output, mascherando le specifiche caratteristiche (fisiche e non) di ogni dispositivo. Il particolare dispositivo *software* che controlla la comunicazione con la periferica è detto anche **Driver** della periferica.

Componenti di un sistema Operativo 2

Gestore della Memoria

Modulo che gestisce la memoria centrale consentendo ai programmi di lavorare in un proprio spazio di indirizzamento virtuale. Permette di

1. proteggere i dati ed i programmi; in generale un programma non può accedere alla zona di memoria riservata ad un altro
2. mascherare l'allocazione fisica dei dati; il programma non vede i dettagli fisici relativi alla collocazione dei dati
3. condividere dati in modo controllato fra più programmi

Nucleo

Componente che comunica con la macchina fisica per gestire lo stato di avanzamento dei processi corrispondenti ai vari programmi che sono contemporaneamente attivi. Il nucleo in sostanza realizza più macchine virtuali corrispondenti all'unica CPU fisica

BIOS (*Basic Input Output System*)

A differenza degli altri componenti, questo modulo, che può essere considerato separato dal Sistema Operativo vero e proprio, non risiede nella memoria di massa ma è memorizzato permanentemente su una ROM (memoria di sola lettura). Al momento dell'accensione l'elaboratore come prima cosa cerca il BIOS che fornisce una sorta di collegamento fra l'*hardware* ed il *software*. Il BIOS gestisce varie funzionalità (gestisce le comunicazioni

attraverso le porte del *computer*, interpreta i dati immessi da tastiera, visualizza i caratteri sullo schermo) in modo tale da rendere il sistema operativo indipendente dall'*hardware*.

Alcuni Sistemi Operativi esistenti

L'interfaccia verso l'utente dell'interprete dei comandi può essere molto diversa:

1. nel caso più semplice (e vecchio) l'interprete dei comandi del sistema operativo si aspetta un comando testuale dal terminale (esempio DOS)
 2. nel caso più moderno l'interfaccia è di tipo testuale (*Windows* di *Microsoft*, *MacOS* di *Apple*, *Linux*): l'interprete dei comandi propone sullo schermo un menu di comandi selezionabili mediante il *mouse*, che permette anche di manipolare icone disposte su una scrivania virtuale (*desktop*).
- nel caso più moderno l'interfaccia è di tipo testuale (*Windows* di *Microsoft*, *MacOS* di *Apple*, *Linux*): l'interprete dei comandi propone sullo schermo un menu di comandi selezionabili mediante il mouse, che permette anche di manipolare icone disposte su una scrivania virtuale (*desktop*).

I sistemi operativi moderni (*Windows*, *MacOS*, *Linux*) permettono il *multitasking*, ovvero permettono l'uso della CPU ad un programma alla volta per brevi intervalli di tempo, così che l'utente vede più programmi eseguiti contemporaneamente. Quando questo non è possibile si parla di sistemi *monotasking*.

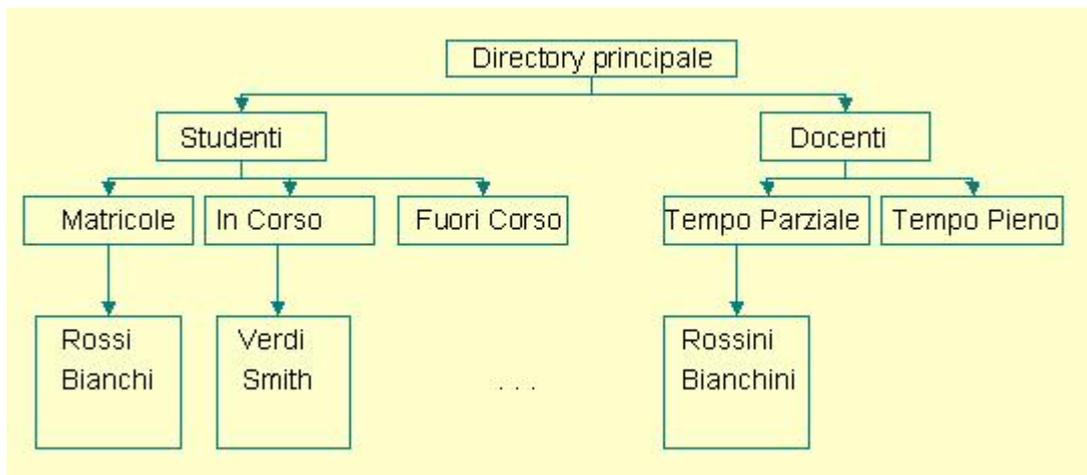
Alcuni sistemi operativi

1. *Unix*: sviluppato a partire dal 1973 presso AT&T. Adatto principalmente per *server* di medie dimensioni.
2. DOS: introdotto nel 1980 sul primo PC IBM, interfaccia testuale, *monotasking*.
3. *MacOS*: sviluppato da *Apple* per i propri *computer* e per PC. *Apple* è stato il primo produttore ad offrire interfacce grafiche (a finestre).
4. *Windows* (95, 98, NT, 2000, ME, CE) *Microsoft*: tipico sistema operativo per PC, interfaccia grafica.
5. *Linux*: basato su *Unix*, sistema operativo per varie piattaforme basato sulla filosofia *freeware*, ossia è distribuito gratuitamente (si paga poi la manutenzione ed i servizi).

Gestione dei file e dei dati 1

I sistemi operativi moderni (*Windows*, *MacOS*, *Linux*) offrono in primo luogo servizi di gestione di documenti di varia natura (testi, grafica, audio, video, programmi). I documenti sono memorizzati in contenitori logici detti *file*, identificati da un nome.

L'insieme dei *file*, detto anche *file system*, è tipicamente organizzato in una struttura ad albero, i cui nodi sono detti *directory* (o cartelle) e le cui foglie contengono i singoli *file*.



Il contenuto di un *file* è di solito memorizzato su un singolo dispositivo di memoria secondaria (disco, CDROM, nastro). L'insieme degli archivi può invece essere distribuito su più dispositivi. Un *file* ha una struttura logica contigua, ma può essere memorizzato su più blocchi di memoria, non necessariamente contigui.

Gestione dei file e dei dati 2

Le varie cartelle sono rappresentate con icone, ad esempio del tipo



La struttura ad albero è rappresentata in vario modo, ad esempio facendo vedere le cartelle interne come raggiungibili dalla cartella esterna cliccando su un simbolo associato alla medesima. Inoltre cliccando su di una cartella di solito si accede al suo contenuto

Ad ogni *file* sono associati alcuni attributi. L'attributo più importante di un *file* è il **nome** che permette l'accesso al *file*. Il nome completo è specificato di solito fornendo un cammino che individua il *file* nella struttura ad albero del *file system*. Ad esempio

C:\didattica\informatica\corso.doc

indica il *file* lucidi nella cartella informatica che a sua volta si trova nella cartella didattica che si trova sull'*hard disk*. Normalmente comunque non serve specificare il nome completo in quanto si possono cercare i *file* aprendo le cartelle fino ad arrivare al *file* desiderato. Per poter accedere al contenuto di un *file* si deve effettuare un'operazione di apertura del *file*, mentre quando si è terminato l'utilizzo di un *file* esso deve essere chiuso. I *file* possono essere spostati da una cartella ad un'altra trascinando le relative icone per mezzo del mouse.

Un altro attributo importante è il **tipo del file**, che descrive la struttura del contenuto. La maggior parte dei sistemi operativi supporta molti tipi distinti di *file*. In DOS e *Windows* il tipo di *file* è indicato da un suffisso che si appende al nome, detto **estensione**. Ad esempio

.txt
denota un *file* in testo ASCII
.doc
denota un *file* scritto con MS *Word*
.htm

denota un *file* HTML

Codice ASCII

Un *file* è rappresentato elettronicamente (su disco, in RAM) come sequenza di caratteri ASCII (ASCII: *American Standard Code for Information Intechange*). L'ASCII standard è su 7 bit, quindi ha 128 valori possibili. Ad esempio

Hex Dec Val

61	97	a
62	98	b
63	99	c
64	100	d
65	101	e
66	102	f
67	103	g

Hex

Valore esadecimale

Dec

Valore decimale

Val

Simbolo rappresentato

Utilità di sistema

Oltre al componente per la gestione di *file*, i sistemi operativi comprendono programmi di utilità (o utility di sistema) che realizzano varie funzionalità utili alla gestione dell'elaboratore. Fra le principali ricordiamo le seguenti:

Formattazione.

Un programma di utilità è adibito alla formattazione dei dischi, che consiste nella suddivisione del disco in tracce circolari e settori a spicchio in modo tale da creare una degli indirizzi che permettano di archiviare e poi recuperare i *file*. Tale programma crea anche la tabella di allocazione dei *file* (FAT) che permette la localizzazione dei *file* fornendo la corrispondenza nome-indirizzo sul disco.

Scanning.

Questo programma di utilità effettua una scansione del disco per individuare eventuali porzioni di memoria rovinate e marcarle in modo tale che non siano poi usate dal sistema operativo. Inoltre permette entro certi limiti di localizzare eventuali dati persi a causa di guasti del sistema.

Eliminazione *file*.

Nei sistemi operativi con interfaccia grafica moderni si possono eliminare i *file* che non servono più spostandoli in un cestino (lo spostamento avviene trascinando le relative icone per mezzo del mouse). Da qui i *file* possono essere recuperati fino a che non si effettua una operazione di svuotamento del cestino (dopo tale operazione il recupero è possibile solo con programmi speciali e se lo spazio disco non è stato sovrascritto).

Utilità di sistema: Deframmentazione e backup

Deframmentazione (*Defrag*).

Quando un disco è vuoto i *file* vengono memorizzati in porzioni di disco (dette cluster) adiacenti. Tuttavia, successive cancellazione e aggiunte di *file* possono rendere sempre più piccoli gli spazi liberi adiacenti, così che un singolo *file* può venire memorizzato su cluster non più adiacenti ma sparsi sul disco. Questo crea un rallentamento delle funzioni del calcolatore perchè i tempi necessari per le operazioni di lettura e scrittura vengono ad essere aumentati. Si può ovviare a questo problema usando una utility di deframmentazione, che sposta i dati sul disco in modo tale da memorizzare i dati di uno stesso *file* su cluster contigui e da ricompattare lo spazio vuoto.

Backup.

Dato che i dischi, come ogni dispositivo fisico, sono soggetti ad errori e malfunzionamenti, per evitare la perdita irrimediabile di dati è utile eseguire periodicamente il salvataggio dei dati di interesse su dispositivi di memoria ausiliari (*hard disk*, *floppy*, DVD o nastri). Tale salvataggio, detto backup, può essere fatto utilizzando un opportuno programma di sistema che di solito provvede anche alla compressione dei dati per ottimizzare l'uso dello spazio di memoria.

L'uso dei programmi di utilità di sistema di solito è semplice. Ad esempio, in *Windows 98*, per accedere alle utilità di sistema basta premere sul bottone *START*, selezionare Programmi (*Programs*) nel menù a tendina che appare, selezionare quindi Accessori (*Accessories*) e quindi Strumenti di Sistema (*System Tools*). Da qui si accede (cliccando sul corrispondente nome) ai vari programmi di utilità il cui uso è autoesplicativo. Nel caso in cui tali programmi non siano raggiungibili dal menu è semplice localizzarli usando il comando *Find* (dal menù a tendina che si ottiene cliccando sul bottone *START*)

Sistemi operativi Stand-alone e di rete

I sistemi operativi si possono distinguere anche in base alla possibilità di supportare applicazioni di rete. Si parla di sistema operativo stand-alone quando il sistema operativo gestisce un unico calcolatore mentre si parla di sistema operativo di rete quando deve gestire dei calcolatori collegati in rete. Uno dei modelli di condivisione delle risorse più comuni prevede un'architettura di tipo *client-server*:

client:

è il calcolatore (o, in generale, il processo) che richiede dei servizi; normalmente i *client* corrispondono ai processi attivati dagli utenti finali;

server:

è un calcolatore che ha il compito di soddisfare le richieste dei *client* fornendo i servizi richiesti e gestendo quindi la condivisione delle risorse.

Ad esempio, nel contesto del *World Wide Web*, i *browser* usati sui *computer* degli utenti che accedono al WWW hanno il ruolo di *client* mentre le macchine sulle quali sono presenti le pagine web sono i *server*.

I sistemi operativi usati sui *server* devono essere in grado di gestire grandi quantità di dati e comunicazioni, devono garantire la sicurezza e la consistenza dei dati memorizzati centralmente sul *server*, devono gestire in modo efficiente la condivisione di risorse. Fra i sistemi operativi più diffusi per l'uso in rete ricordiamo:

1. *Windows NT*
2. *OS 2*
3. *Unix*
4. *Linux*

Condivisione e trasferimento di file

Un'applicazione molto comune in una rete di calcolatori è il trasferimento di *file* fra *computer* diversi.

1. Si dice caricare un *file* (o farne l'*upload*) il trasferirlo ad un altro calcolatore
2. Si dice scaricare un *file* (o farne il *download*) il trasferirlo da un altro calcolatore al nostro.

Se si sta operando all'interno di una rete locale (LAN) di calcolatori il trasferimento di un *file* avviene essenzialmente come il trasferimento fra due cartelle diverse all'interno della stessa macchina: basta spostare il *file* in una cartella pubblica, ovvero una cartella alla quale abbiano accesso anche altri utenti: normalmente al momento della creazione di una cartella è possibile stabilirne i diritti di accesso e quindi se la cartella è condivisa (*shared*) o no. In *Windows*, ad esempio, si possono modificare i diritti d'accesso cliccando con il bottone destro del mouse sulla cartella e quindi selezionando proprietà (*properties*) e poi condivisione (*sharing*).

Se invece si vogliono trasferire *file* fra *computer* geograficamente distanti ma connessi ad Internet (o comunque connessi da un rete geografica) si può usare il protocollo FTP. Questo è un protocollo del livello applicazione che usa TCP come protocollo di livello trasporto. Se dunque è disponibile FTP (e quindi TCP) si possono scambiare *file* mediante una sorta di *login* remoto e quindi usando i comandi di FTP.

Una terza possibilità infine è quella di inviare un *file* come *attachment* di un messaggio di posta elettronica. In questo caso si usano opportuni programmi di codifica per evitare che nel trasferimento del messaggio attraverso i vari *computer* (*gateway*) sulla rete si generino errori. La codifica è solitamente fatta automaticamente dal programma che si usa per la posta elettronica. Ad esempio, l'estensione MIME del protocollo SMTP utilizza un sistema di codifica detto Base 64 e trasforma un *file* binario in un *file* ASCII.

FTP

Ad esempio, per trasferire il *file* *pluto* dalla macchina *pippo.cs.psu.edu* alla propria macchina si può fare come segue:

1. si digita
`ftp pippo.cs.psu.edu`
2. nella finestra in cui è disponibile una *shell* (in *Unix* o *Linux*) o nella zona opportuna della finestra ottenuta cliccando sul bottone *START* e quindi selezionando *Esegui* (*Run*) (in *Windows*). Questo fa sì che venga eseguito il programma FTP con accesso alla macchina *pippo.cs.psu.edu*.
3. si effettua il *login* fornendo nome utente e *password*. Se non si dispone di un *account* sulla macchina *pippo* si può usare la parola *anonymous* come nome utente e quindi il proprio indirizzo di *e-mail* come *password*.
4. si digita
`get pluto` per trasferire il *file*
5. si digita
`quit` per terminare il collegamento quando la trasmissione è finita.

Se il *file* richiesto non fosse al *top-level* nella macchina alla quale ci si è collegati, si possono usare opportuni comandi di FTP, simili a quelli di *Unix*, per muoversi nel *file system* e localizzarlo. È possibile anche ottenere aiuto da FTP digitando *help* oppure il punto interrogativo.