**Parte II - Capitolo 1 - Introduzione**

Un ***sistema operativo*** è un insieme di programmi che gestisce gli elementi fisici di un calcolatore, fornisce una piattaforma ai programmi applicativi e agisce da intermediario fra l'utente e la struttura fisica del calcolatore. Un sistema di calcolo è formato da quattro componenti: *dispositivi fisici*, *sistema operativo*, *programmi applicativi* e *utenti*. Il sistema operativo ha il compito di controllare e coordinare l'uso dei dispositivi fisici (CPU, memoria e dispositivi di I/O) da parte dei programmi applicativi (editor di testo, fogli di calcolo, compilatori) per gli utenti. La percezione dell'utente dipende principalmente dall'interfaccia impiegata. Il sistema operativo si deve progettare considerando principalmente la *facilità d'uso*, con qualche attenzione alle prestazioni. Alcuni utenti possono usare terminali connessi a *mainframe* o *minicalcolatori* condividendone le risorse con altri utenti connessi a terminale. In questo caso il sistema operativo si progetta per massimizzare l'utilizzo delle risorse garantendo che i dispositivi fisici siano impiegati in modo equo ed efficiente. Negli ultimi anni si sono diffusi i calcolatori palmari nei quali il sistema operativo ha il compito di facilitare l'uso individuale e ridurre il consumo delle batterie. Dal punto di vista del calcolatore, invece, il sistema operativo è il programma più strettamente collegato ai suoi elementi fisici. È possibile quindi considerare un sistema operativo come un *assegnatore di risorse*. Il sistema operativo è un *programma di controllo* che gestisce l'esecuzione dei programmi utente in modo da impedire che si verifichino errori o che il calcolatore sia usato in modo scorretto. Un modo comune di definire un sistema operativo è quella secondo cui quest'ultimo è il solo programma che funziona sempre nel calcolatore ed è generalmente chiamato ***kernel*** (***nucleo***).

Vediamo ora come funziona un sistema di calcolo moderno. L'avviamento del sistema, conseguente all'accensione fisica di un calcolatore, richiede la presenza di uno specifico programma iniziale, di solito non troppo complesso, detto ***programma di avviamento*** (***bootstrap program***) in genere contenuto in tipi di memoria noti con il termine generale di ***firmware***, il cui supporto fisico è parte integrante della macchina. La funzione del programma di avviamento è quella di inizializzare i diversi componenti del sistema (registri, controllori, contenuto della memoria centrale); deve inoltre caricare in memoria il sistema operativo e avviarne l'esecuzione, individuando e caricando in memoria il kernel del sistema operativo. Il sistema operativo avvia l'esecuzione del primo processo d'elaborazione e attende il verificarsi di un evento. Un evento è di solito segnalato da un'interruzione dell'attuale sequenza di esecuzione della CPU, che può essere causata da un dispositivo fisico (segnale di interruzione) o da un programma (interrupt). Un segnale d'interruzione deve trasferire il controllo all'appropiata procedura di servizio dell'evento a esso associato. La gestione delle interruzioni deve essere molto rapida, quindi ci si serve di una tabella di puntatori alle specifiche procedure, dato che i segnali di interruzione sono predefiniti. Questa sequenza di indirizzi è detta ***vettore delle istruzioni***. L'architettura di gestione delle interruzioni deve salvare l'indirizzo dell'istruzione interrotta nello stack di sistema. Terminato il servizio dell'interruzione, l'indirizzo di ritorno viene caricato nel ***program counter*** che contiene l'indirizzo della prossima istruzione da eseguire, in modo da riprendere l'esecuzione come se non si fosse verificata nessuna eccezione.

Una percentuale cospicua del codice di un sistema operativo è dedicata alla gestione dell'I/O poichè i dispositivi proposti in ingresso e in uscita sono molto diversi tra loro. Un calcolatore d'uso generale è composto da una CPU e da un insieme di controllori di dispositivi connessi tramite bus comune. Un controllore SCSI (*small computer-system interface*) è capace di controllare sette o più dispositivi. Un controllore di dispositivo contiene una memoria interna detta *memoria di transito* (*buffer*) e un insieme di registri specializzati. Il sistema operativo possiede un ***driver del dispositivo*** che si coordina con il controllore e funge da interfaccia uniforme con il resto del sistema. Per avviare un'operazione di I/O, il driver del dispositivo carica i registri interessati all'interno del controllore che esamina i contenuti di questi registri per scegliere l'azione da intraprendere. Il controllore inizia a trasferire i dati dal dispositivo al proprio buffer locale. Quando il trasferimento è completato, il controllore informa il driver tramite un'interruzione e il controllo viene passato al sistema operativo. Questo metodo non è adatto a trasferimenti massicci di dati, quindi si ricorre alla tecnica di ***accesso diretto alla memoria*** (***DMA***) che consiste nell'impostare i buffer, i puntatori e i contatori necessari al dispositivo di I/O, dopodichè il controllore trasferisce un intero blocco di dati dalla propria memoria di buffer direttamente alla memoria central, senza l'intervento della CPU. In questo modo l'operazione richiede una sola interruzione e la CPU resta libera per svolgere altri compiti.

Fin ora è stato presentata la struttura generale di un calcolatore, ma questo può essere organizzato secondo criteri molto diversi. Un ***sistema monoprocessore*** è dotato di una CPU principale in grado di eseguire un insieme di istruzioni di natura generale, comprese quelle necessarie ai processi utenti. Quasi tutti i sistemi possiedono altri processori specializzati in compiti particolari. Questi processori particolari hanno un insieme ristretto di istruzioni che possono eseguire e non possono svolgere i processi utenti. L'utilizzo di microprocessori con finalità specifiche non trasforma un sistema monoprocessore in un sistema multiprocessore. Un ***sistema multiprocessore***, conosciuto anche con il nome di *sistema parallelo* o *sistemi strettamente connessi* (*tightly coupled system*), dispone di più unità di elaborazione in stretta comunicazione, che condividono i canali di comunicazione all'interno del calcolatore (bus), i timer dei cicli di macchina (clock) e talvolta i dispositivi di memorizzazione e periferici. I vantaggi principali prodotti da tali sistemi sono: maggiore produttività (throughput), cioè aumentando il numero di unità di elaborazione è possibile svolgere un lavoro maggiore in meno tempo; economia di scala, cioè se più programmi devono operare sullo stesso insieme di dati, è economicamente più conveniente registrarli in dischi condivisi da tutte le unità di elaborazione; incremento dell'affidabilità, cioè se le funzioni sono distribuite adeguatamente, un guasto di un'unità di elaborazione non blocca il sistema, ma semplicemente lo rallenta. La capacità di continuare a offrire un servizio ai dispositivi ancora in funzione è detta ***degradazione controllata*** (***graceful degradation***); i sistemi con questa capacità sono detti ***tolleranti ai guasti*** (***fault-tolerant***) e necessitano di un meccanismo per il riconoscimento, la diagnosi e la riparazione del danno quando è possibile. I multiprocessori attualmente sono di due tipi: alcuni impiegano la ***multielaborazione asimmetrica*** (***asymmetric multiprocessing***, ***AMP***), in cui a ogni unità di elaborazione si assegna un compito specifico, il tutto controllato da un'unità di elaborazione principale; altre invece impiegano la ***multielaborazione simmetrica*** (***symmetric multiprocessing***, ***SMP***), in cui ogni processore è abilitato al compimento di tutte le operazioni del sistema. La differenza principale tra la multielaborazione simmetrica e asimmetrica può derivare da caratteristiche architetturali e del sistema operativo. La multielaborazione può causare il cambiamento del modello di accesso alla memoria, trasformando un accesso uniforme (UMA), cioè un accesso alla RAM tramite qualsiasi CPU richiede lo stesso tempo, in un accesso non uniforme alla memoria (NUMA), cioè l'accesso ad alcune parti della memoria richiede più tempo, quindi c'è un peggioramento delle prestazioni. Una tendenza recente nella progettazione della CPU è quella di raggruppare diverse unità di calcolo (*core*) in un singolo circuito. Un ***cluster*** o cluster di elaboratori (*clustered system*) è basato sull'uso congiunto di più unità di elaborazione riunite per lo svolgimento di attività d'elaborazione comuni. Differiscono dai sistemi paralleli perchè comprendono due o più calcolatori completi collegati tra loro. Questo tipo di soluzione viene adottata per offrire un'elevata disponibilità. Ciascun calcolatore esegue una serie di programmi che forma uno strato di gestione del cluster; ogni nodo può tenere sotto controllo tramite LAN uno o più degli altri nodi. I cluster sono strutturabili in modo simmetrico e asimmetrico. Nei ***cluster asimmetrici*** un calcolatore rimane nello stato di *attesa attiva* (*hot standby mode*) mentre l'altro esegue le applicazioni. Nei ***cluster simmetrici*** due o più calcolatori eseguono le applicazioni e allo stesso tempo si controllano reciprocamente ottenendo maggiore efficienza. I cluster possono anche essere usati per ottenere ambienti di elaborazione ad alte prestazioni (*high-performance computing*); sistemi di questo tipo offrono molta più potenza di calcolo, ma le applicazioni devono essere scritte in modo da trarre vantaggio dal cluster sfruttando una tecnica chiamata ***parallelizzazione*** (***parallelization***) che consiste nel suddividere il programma in componenti separate, eseguibili in parallelo su singoli computer all'interno del cluster.

Fra le più importanti caratteristichè dei sistemi operativi vi è la ***multiprogrammazione*** che consente di aumentare la percentuale di utilizzo della CPU, organizzando il lavoro in modo tale da mantenerla in continua attività. Il sistema operativo tiene contemporaneamente in memoria centrale diversi lavori; dato che, in genere, la memoria centrale è troppo piccola per contenere tutti i programmi da eseguire, questi vengono collocati inizialmente sul disco in un'area apposita detta ***job pool*** che contiene tutti i programmi in attesa di essere allocati nella memoria centrale. Il sistema operativo sceglie un lavoro da compiere e, se questo si trova in attesa di qualche evento, passa a un altro lavoro e lo esegue (in un sistema non multiprogrammato la CPU rimarrebbe inattiva). La ***partizione del tempo d'elaborazione*** (*time sharing* o *multitasking*) è un'estensione logica della multiprogrammazione: la CPU esegue più lavori commutando le loro esecuzioni con frequenza tale da permettere a ciascun utente l'interazione con il proprio programma durante la sua esecuzione. Un ***sistema di calcolo interattivo*** permette la comunicazione diretta tra utente e sistema. L'utente impartisce le istruzioni direttamente al sistema, quindi il tempo di risposta dovrebbe essere breve. Un programma caricato in memoria e predisposto per l'esecuzione è noto come ***processo***. Se alcuni processi sono pronti per il trasferimento in memoria centrale, ma lo spazio disponibile non è sufficiente per accoglierli tutti, il sistema deve fare una selezione che va sotto il nome di ***job scheduling*** o *pianificazione dei lavori*. I problemi inerenti alla scelta dei processi da eseguire vengono affrontati con lo ***scheduling della CPU***. In sistemi basati sulla partizione di tempo di elaborazione, il sistema operativo deve garantire tempi di risposta accettabili: questa finalità è raggiunta grazie alla tecnica dello ***swapping*** che consente di scambiare i processi presenti in memoria con quelli che risiedono sul disco e viceversa. Un metodo più comune per ottenere lo stesso risultato è la ***memoria virtuale*** che consente l'esecuzione di lavori d'elaborazione anche non internamente caricati nella memoria.

Per definire un corretto funzionamento del sistema, è necessario distinguere due diverse modalità di esecuzione: ***modalità utente*** e ***modalità di sistema*** (detta anche *modalità kernel***,** *modalità supervisore*, *modalità monitor* o *modalità privilegiata*). Per indicare quale modalità è attiva, l'architettura della CPU deve essere dotata di un ***bit di modalità*** (0 di sistema, 1 utente) che consente di stabilire se l'istruzione corrente è eseguita per conto del sistema operativo o per conto dell'utente. La duplice modalità di funzionamento (*dual mode*) consente la protezione del sistema operativo rispetto al comportamento degli utenti e viceversa. Questo livello di protezione si ottiene definendo come ***istruzioni privilegiate*** le istruzioni di macchina che possono causare danni allo stato del sistema. L'esecuzione di queste istruzioni possono essere eseguite solo in modalità sistema (in caso contrario la CPU non esegue l'istruzione e invia un'eccezione al sistema operativo). Le chiamate di sistema sono strumenti con cui un programma utente richiede al sistema operativo di compiere operazioni a esso riservate per conto del programma utente. Queste sono realizzate tramite eccezioni che rimandano a un indirizzo specifico nel vettore delle interruzioni. (*trap* generica o *syscall*). Se la CPU non dispone delle due modalità di funzionamento il sistema operativo può andare incontro a serie limitazioni (un programma utente potrebbe cancellare il sistema operativo scrivendo nuove informazioni nelle locazioni in cui esso è memorizzato). Occorre dunque assicurare che il sistema operativo mantenga il controllo dell'elaborazione, cioè impedire che un programma utente entri in ciclo infinito o non richieda servizi al sistema. A tale scopo viene utilizzato un ***timer variabile*** di solito realizzato mediante un generatore di impulsi a frequenza fissa e un contatore, che invia un segnale di interruzione alla CPU a intervalli di tempo specificati. La presenza di un timer garantisce che un programma utente non possa essere eseguito troppo a lungo.

Un processo è un'entità attiva, con un contatore di programma che indica la successiva istruzione da eseguire. Il sistema operativo è responsabile delle seguenti attività per quanto riguarda la gestione dei processi: creazione e cancellazione dei processi utenti e sistema, sospensione e ripristino dei processi, fornitura di meccanismi per la sincronizzazione dei processi, per la comunicazione tra i processi, per la gestione dei deadlock.

Il sistema operativo è responsabile delle seguenti attività connesse alla gestione della memoria centrale: tenere traccia di quali parti di memoria sono attualmente usati e da chi, decidere quali processi e dati debbano essere caricati in memoria o trasferiti altrove, assegnare e revocare lo spazio di memoria secondo le necessità.

Per facilitare gli utenti, il sistema operativo fornisce un'interfaccia logica per la memorizzazione delle informazioni, definendo un'unità logica di archiviazioni: il ***file***. La gestione del file è uno dei componenti più visibili di un sistema operativo. Un file è una raccolta di informazioni correlate definite dal loro creatore. Il sistema operativo realizza il concetto astratto di file gestendo i mezzi di memoria di massa e i dispositivi che li controllano. Il sistema operativo è responsabile delle seguenti attività connesse alla gestione dei file: creazione e cancellazione dei file, creazione e cancellazione di directory, fornitura delle funzioni fondamentali per la gestione di file e directory, associazione dei file ai dispositivi di memoria secondaria, creazione di copie di riserva (*backup*) dei file su dispositivi di memorizzazione non volatili.

Dato che la memoria centrale è troppo piccola per contenere tutti i dati e tutti i programmi, e il suo contenuto va perduto se il sistema si spegne, il calcolatore deve disporre di una memoria secondaria a sostegno della memoria centrale. La maggior parte dei calcolatori moderni utilizza i dischi per la memorizzazione secondaria. Il sistema operativo è responsabile delle seguenti attività connesse alla gestione dei dischi: gestione dello spazio libero, assegnazione dello spazio, scheduling del disco. Vi sono svariati frangenti in cui tornano utili dispositivi di memorizzazione più lenti, meno costosi e più capienti dei dispositivi di memoria secondaria. Sono i dispositivi di memoria terziaria che comprendono le unità a nastro magnetico, i CD e i DVD, con diversi formati (WORM, write once read many; RW, read and write).

Il sistema operativo deve garantire la protezione del sistema evitando la collisione tra i programmi. Per ***protezione*** si intende ciascuna strategia di controllo dell'accesso alle risorse possedute da un elaboratore, da parte di processi o utente. Le strategie di protezione devono fornire le specifiche dei controlli da attuare e gli strumenti per la loro effettiva applicazione. Pur essendo dotato di protezione adeguata, il sistema può rimanere esposto agli eccessi abusivi, rischiando malfunzionamenti. È compito della ***sicurezza*** difendere il sistema dagli attacchi provenienti dall'interno o dall'esterno. La varietà di minacce conosciute è enorme: si va dai virus e i worm, agli attacchi che paralizzano il servizio, al furto di identità e la sottrazione del servizio. La protezione e la sicurezza presuppongono che il sistema sia in grado di distinguere tra tutti i propri utenti. Nella maggior parte dei sistemi operativi è disponibile un elenco di nomi degli utenti e dei loro ***identificatori utente*** (***user ID***) che sono id numerici che identificano univocamente l'utente.

Per ***sistema distribuito*** si intende un insieme di elaboratori collocati a distanza, e con caratteristiche spesso eterogenee, interconnessi da una rete di calcolatori per consentire agli utenti l'accesso alle varie risorse dei singoli sistemi. L'accesso a una risorsa condivisa in rete aumenta la velocità di calcolo, la funzionalità, la disponibilità dei dati e il grado di affidabilità. Una ***rete*** si può considerare come un canale di comunicazione tra due o più sistemi. I sistemi distribuiti si basano sulle reti per realizzare le proprie funzioni. Le reti differiscono tra loro per i protocolli usati (il più diffuso è il TCP/IP) e possono essere classificate secondo le distanze tra i loro nodi: una rete locale (***LAN***) comprende nodi all'interno della stessa stanza, piano o edificio; una rete geografica (***WAN***) si estende a gruppi di edifici, città, al territorio di una regione o uno stato; le reti metropolitane (***MAN***) collegano gli edifici di un'intera città; i dispositivi che si collegano a breve distanza, tramite per esempio BlueTooth, creano una ***microrete*** (***small-area network***) simile a quelle domestiche. I mezzi di trasmissione che si impegnano nelle reti sono altrettanto vari: fili di rame, fibre ottiche, trasmissioni via satellite, sistemi a microonde e sistemi radio. Un ***sistema operativo di rete*** è dotato di caratteristiche quali la condivisione dei file attraverso la rete e di un modello di comunicazione per i processi attivi su elaboratori diversi, che possono così scambiarsi messaggi. Un computer che funzioni con un sistema operativo di rete agisce con una certa autonomia rispetto a tutti gli altri computer della rete.

Fino a questo momento sono stati trattati sistemi di calcolo con finalità generali. Consideriamo ora alcuni sistemi di calcolo le cui funzioni sono più limitate. I ***computer integrati*** (***embedded computers***) costituiscono la tipologia predominante di elaboratore. Sono i dispositivi che si trovano nei motori delle auto, nei robot industriali, nei videoregistratori e nei forni a microonde. Hanno compiti molto precisi e un'interfaccia utente scarsamente sviluppata, se non del tutto assente. Questi funzionano quasi sempre con *sistemi operativi real-time* che vengono utilizzati quando sono stati imposti rigidi vincoli di tempo a carico del processore o al flusso dei dati. L'elaborazione deve avvenire nei vincoli di tempo definiti in modo che le prestazioni siano soddisfacenti e producano un risultato corretto.

Per effetto delle tendenze di mercato, hanno fatto la loro comparsa i ***sistemi multimediali*** che hanno il compito di riprodurre i dati multimediali o a trasmissione continua, garantendo precise frequenze.

I ***sistemi palmari*** (***handheld systems***) comprendono gli assistenti digitali noti come PDA, i computer tascabili e i telefoni cellulari, i quali adottano sistemi operativi integrati a orientamento specifico.

Con l'evoluzione delle tecniche di elaborazione, i confini tra molti ambienti d'elaborazione tradizionale sono diventati sempre più sfumati. Grazie alle connessioni veloci è possibile avere un ambiente di elaborazione in modo semplice. Alcuni ambienti di elaborazione domestici sono dotati di ***firewall*** (***barriere antintrusione***) che proteggono dagli attacchi informatici esterni. Con l'aumento della velocità, della potenza e dell'economicità dei PC, i progettisti si sono diretti verso l'utilizzo dei sistemi come sistemi server per le richieste che vengono ricevute dai sistemi client. I sistemi server si possono suddividere in ***server per l'elaborazione***, che forniscono un'interfaccia a cui il client può inviare una richiesta, e ***file server***, che offrono un'interfaccia al file system che consente al client creazione, aggiornamento, lettura e cancellazione dei file. Un'altra architettura di sistemi distribuiti è il modello di sistema da pari a pari, o ***peer-to-peer*** (***P2P***). In questo modello cade la distinzione tra client e server: tutti i nodi all'interno del sistema sono su un piano di parità e ciascuno può fungere ora da client, ora da server, a seconda che stia richiedendo o fornendo un servizio. Esistono due modalità generali per stabilire quali servizi siano disponibili: al momento di unirsi a una rete, un nodo iscrive il proprio servizio in un registro centralizzato di consultazione della rete; una risorsa di rete che operi in veste di client deve accertare quale nodo fornisca il servizio desiderato, e lo fa inoltrando la richiesta di servizio a tutti gli altri nodi della rete (*protocollo di scoperta*)

I ***sistemi operativi open-source*** sono disponibili in formato sorgente anzichè come codice binario compilato. Linux è il più diffuso sistema operativo open-source, mentre Microsoft Windows è un ben noto esempio dell'approccio opposto, ***a sorgente chiuso*** (***closed source***). Avere a disposizione il codice sorgente permette al programmatore di produrre codice binario eseguibile dal sistema. Il processo inverso, chiamato ***reverse-engineering*** permette di ricavare il codice sorgente partendo dal binario; questo processo risulta molto più oneroso. Il vantaggi dei sistemi operativi open source è la presenza di programmatori interessati che contribuiscono allo sviluppo aiutando a verificare la presenza di eventuali errori nel codice. (+ pag 37)