

SISTEMI OPERATIVI

Un sistema operativo è un programma che costituisce l'interfaccia tra utente e hardware e permette all'utente l'utilizzo dell'elaboratore tramite il linguaggio di controllo; stabilisce le modalità di funzionamento; consente l'ottimizzazione delle risorse disponibili; per risorse si intendono tutte le componenti hardware e software: CPU, memoria, periferiche, informazioni. Insieme all'hardware costituisce la macchina estesa cioè la macchina con cui l'utente interagisce. Esistono vari tipi di sistemi operativi che permettono di svolgere funzioni diverse e di operare con modalità diverse. Uno stesso computer può funzionare in modo diverso con sistemi operativi diversi (devono comunque essere disponibili le opportune apparecchiature hardware).

STRUTTURA DEL SISTEMA OPERATIVO

Il sistema operativo può essere suddiviso in modo gerarchico in livelli ognuno dei quali opera utilizzando routine dei livelli inferiori. Si hanno allora partendo dal livello inferiore:

- il nucleo (o supervisore) che comprende le procedure di gestione della CPU e la gestione delle interruzioni;

- il gestore della memoria;

- il gestore delle periferiche;

- il gestore delle informazioni;

- il modulo (shell) che permette di interpretare i comandi del linguaggio di controllo con cui si può operare, i programmi di utilità (editor, compilatori, interpreti, linker, caricatori, programmi di gestione file etc.)

MODALITÀ DI FUNZIONAMENTO

La macchina estesa (hardware + sistema operativo) stabilisce le modalità con cui si può operare. Dal punto di vista della gestione dei dati si può operare in modo batch o interattivo. In base all'elaborazione dei programmi le modalità di funzionamento del sistema operativo sono:

- uniprogrammazione,

- multiprogrammazione,

- time-sharing.

UNIPROGRAMMAZIONE

In uniprogrammazione può essere caricato in memoria un solo programma alla volta. In modo interattivo un solo utente può lavorare; un programma batch viene eseguito immediatamente; più programmi vengono eseguiti in sequenza uno dopo l'altro.

Il sistema più noto di computer in uniprogrammazione è il personal computer.

Sul personal in generale oltre alla CPU sono presenti più processori:

- per la gestione del video,

- per la gestione del driver,

- per la gestione della stampante.

Alcune operazioni possono essere fatte "contemporaneamente" all'elaborazione: rinfresco del video, immissione di un comando mentre terminano operazioni sul drive. Il sistema operativo offre servizi di:

- controllo programma: avvio, allocazione, esecuzione, terminazione etc.;

- gestione operazioni I/O tramite i moduli device driver per la gestione delle periferiche;

- gestione file: visione logica del disco e accesso ai record del file tramite il file system;

- gestione errori.

MULTIPROGRAMMAZIONE

Nei sistemi di multiprogrammazione più programmi possono essere caricati contemporaneamente in memoria.

La CPU può eseguire soltanto un programma alla volta, ma quando si verifica una interruzione e il programma resta in attesa del completamento di un servizio, la CPU può eseguire le istruzioni di un altro programma.

I programmi presenti in memoria vengono eseguiti alternativamente dalla CPU che può quindi essere sfruttata maggiormente (resta inattiva per periodi minori).

I programmi eseguibili in multiprogrammazione sono programmi di tipo batch: infatti non è possibile determinare il tempo di risposta di un programma.

I programmi vengono inseriti in code di attesa, prelevati dallo schedatore dei lavori (una routine del sistema operativo) e il loro avanzamento dipende dalle interruzioni che si verificano.

L'esecuzione di un programma viene chiamata processo; il programma costituisce la descrizione delle istruzioni da eseguire mentre il processo è la sequenza di azioni compiute durante l'esecuzione.

Quando si richiede l'esecuzione di un programma viene creato per tale programma un descrittore (JCB: Job Control Block), cioè una struttura di dati in cui vengono inserite tutte le informazioni relative al programma:

le risorse necessarie per l'esecuzione,
la priorità del lavoro etc.

Alcune di queste informazioni sono inserite direttamente dall'utente tramite il linguaggio di controllo, altre sono ricavate e gestite solo dal sistema operativo.

Un'apposita routine del sistema operativo inserisce tutte le informazioni necessarie nel JCB e lo mette nella lista dei programmi in attesa di essere caricati in memoria (programmi in stato di disponibilità).

Tutti i descrittori dei programmi che si trovano nello stesso stato sono collegati tra loro in strutture a liste. In stato di disponibilità il programma si trova in memoria di massa; nessuna risorsa è stata ancora associata al programma; viene solamente preparato l'elenco delle risorse necessarie.

Lo schedatore dei lavori sceglie tra i lavori in stato di disponibilità quale portare in memoria (il programma passa così in stato di pronto) basandosi su una strategia detta "politica di schedulazione".

La politica di schedulazione può essere basata sull'ordine di arrivo dei lavori (FIFO) o sulla priorità del lavoro, calcolata dal responsabile del centro o in modo automatico in base alle risorse richieste dal lavoro (priorità statica).

La priorità può anche essere incrementata in relazione al tempo di attesa in coda (priorità dinamica) per evitare che un lavoro possa restare all'infinito in coda di attesa se ci sono sempre lavori a priorità più alta.

Un programma passa dallo stato di disponibilità allo stato di pronto quando gli vengono allocate tutte le risorse necessarie (tranne la CPU).

Il programma viene caricato nella memoria centrale e viene creato un nuovo descrittore contenente tutte le informazioni relative al processo creato (PCB: Process Control Block).

Il Process Control Block contiene le informazioni relative all'esecuzione del processo:

valori del registro di stato e dei registri di lavoro, (attivo, pronto ecc.) e quelle relative alle risorse associate al processo:

informazioni sulle aree di memoria in cui è memorizzato il processo, sulle periferiche associate e i file aperti, più tutte le informazioni che il sistema operativo usa per effettuare la schedulazione dei processi da assegnare alla CPU: priorità, tempo di CPU utilizzato ecc.

Lo schedatore dei processi sceglie tra quelli in stato di pronto a quale assegnare la CPU: il processo passa allora allo stato di esecuzione.

Un solo programma alla volta può trovarsi in stato di esecuzione.

Lo schedatore dei processi sceglie tra i processi in stato di pronto basandosi su un algoritmo di schedulazione predefinito.

Dallo stato di esecuzione il processo può passare in stato di attesa quando non può avanzare perché aspetta il completamento di una operazione (per esempio di I/O).

In tal caso viene rilasciata la CPU che può essere assegnata ad un altro processo. Quando l'operazione richiesta è conclusa il processo in stato di attesa viene portato nuovamente in stato di pronto.

Se durante l'esecuzione il processo giunge a conclusione passa allo stato di terminazione in cui vengono rilasciate le risorse ed eliminato il PCB relativo.

Il processo può anche passare dallo stato di esecuzione allo stato di pronto se si verifica una interruzione esterna.

TIME SHARING

Caratteristica dei sistemi in time-sharing è il tipo di lavoro interattivo. Non si parla in tal caso di processi ma di sessioni di lavoro in cui più utenti sono collegati tramite terminale e possono eseguire operazioni di diverso tipo.

La sessione di lavoro inizia con il collegamento dell'utente (log in) e termina con lo scollegamento del terminale (log off).

Durante la sessione si possono richiamare comandi del sistema operativo, utilizzare programmi di utilità come editor, compilatori ecc., eseguire programmi applicativi, di solito di tipo interattivo ecc.

La CPU viene assegnata alternativamente alle diverse sessioni di lavoro per un certo periodo di tempo. In questo modo ogni utente ha l'impressione di avere sempre la CPU a sua disposizione.

Il time-sharing è gestito da un programma di sistema detto monitor del time-sharing.

I processi attraversano gli stessi stati di avanzamento descritti per l'elaborazione in multiprogrammazione ma un programma può anche passare dallo stato di esecuzione allo stato di pronto perché è terminato l'intervallo di tempo concesso.

Inoltre poiché non è possibile attendere che i lavori presenti in memoria siano terminati per accettare la richiesta di esecuzione di un programma interattivo da parte di un utente, il sistema operativo deve poter far spazio nella memoria riportando un programma dallo stato di pronto o di attesa allo stato di disponibilità (riportandolo quindi in memoria di massa); questo procedimento viene chiamato swapping.

Se viene richiesta una mole di lavori troppo grossa per le dimensioni del sistema, il tempo di risposta aumenta considerevolmente; infatti la CPU viene impiegata a lungo per l'esecuzione di istruzioni del sistema operativo e per la gestione dello swapping.

TEMPO REALE

Si parla di elaborazione in tempo reale quando i tempi di risposta sono compatibili con il tipo di programma che viene eseguito. La CPU non viene suddivisa tra i programmi a priori dal sistema operativo ma il passaggio della CPU da un programma all'altro è guidato attraverso meccanismi di interrupt dagli eventi che si verificano.

L'elaborazione in tempo reale viene utilizzata per esempio per il controllo di processi industriali in cui il verificarsi di una condizione anomala deve immediatamente essere gestito dal programma. Le elaborazioni in tempo reale di solito utilizzano sistemi dedicati.

SERVIZI DEL SISTEMA OPERATIVO

La possibilità di gestire la multiprogrammazione o il time-sharing dipende essenzialmente dal sistema operativo, anche se, naturalmente, deve essere disponibile la configurazione hardware necessaria.

Il sistema operativo diventa molto complesso poiché deve gestire l'interazione tra diversi processi o utenti; deve perciò svolgere anche funzioni di allocazione delle risorse, gestione dei conflitti che si creano dalla richiesta contemporanea di risorse e protezione delle risorse stesse.

ALLOCAZIONE E GESTIONE DELLE RISORSE

Il sistema operativo in base alle richieste dei processi stabilisce l'assegnazione delle singole risorse. Il sistema operativo costituisce un'interfaccia tra le risorse stesse e i processi; un processo cioè utilizza le risorse senza che debbano essere specificati tutti i dettagli tecnici derivanti dalla implementazione della risorsa stessa; è il sistema operativo che si occupa della gestione della risorsa in un modo che rimane trasparente all'utente e gli consente di preoccuparsi solo delle operazioni logiche non di come queste debbano poi essere realizzate in pratica.

Il sistema operativo comprende dei moduli per la gestione delle risorse (gestione della CPU, della memoria, delle periferiche, delle informazioni) ognuno dei quali deve stabilire a quale processo, tra quelli che la richiedono, assegnare la risorsa, procedere all'assegnazione vera e propria e al termine gestire il rilascio della stessa da parte del processo che l'ha utilizzata.

Per compiere queste operazioni devono essere mantenute aggiornate delle tabelle con le informazioni sullo stato di ciascuna risorsa: se è libera o occupata, a quale processo è assegnata etc.

Le risorse possono essere allocate staticamente, cioè assegnate per tutta la durata dell'esecuzione, o dinamicamente cioè assegnate di volta in volta solo quando necessario e rilasciate subito dopo.

Alcune tecniche di allocazione delle risorse sono più elementari, vengono implementate in modo abbastanza semplice ma non consentono un alto rendimento; al contrario le tecniche più sofisticate permettono una maggior efficienza ma risultano più difficili da realizzare.

Bisogna comunque tenere presente che l'efficienza di un sistema viene misurata in base al rapporto tra il tempo in cui la CPU è attiva e quello in cui la CPU è inattiva, poiché non può essere a nessun processo. Per migliorare l'efficienza complessiva possono peggiorare le prestazioni relative ad un singolo processo cioè può aumentare il tempo di risposta (tra la richiesta di esecuzione e l'emissione del risultato) e il tempo di esecuzione (tra l'inizio e la fine dell'esecuzione) del processo stesso. La politica di gestione delle risorse non deve comunque causare un sovraccarico di lavoro della CPU costringendo la CPU stessa ad eseguire per troppo tempo routine del sistema operativo a scapito dei programmi utente (overhead del sistema)

COMPETIZIONE TRA PROCESSI

In un sistema multiprogrammato, più processi possono competere per l'uso di una risorsa; ogni risorsa può essere assegnata ad un solo processo alla volta e prima o poi ogni processo che la richiede deve riuscire ad ottenerla (problema della mutua esclusione).

Si dice che più processi interferiscono tra di loro se dipendono uno dall'altro per l'uso di una risorsa, ma sarebbero altrimenti indipendenti.

Ad ogni risorsa sono associati dei flag (semaforo) che segnalano se la risorsa è disponibile o occupata.

Quando un processo richiede una risorsa viene esaminato lo stato del semaforo; se la risorsa è libera si pone il flag in stato di occupato e si assegna la risorsa al processo; quando il processo ha terminato di usarla la rilascia ponendo di nuovo il flag in stato disponibilità.

Se invece al momento della richiesta la risorsa è già occupata, il processo viene posto in stato di attesa in una coda di attesa per la risorsa richiesta, da cui potrà essere prelevato solo quando la risorsa risulterà libera.

L'accesso al semaforo della risorsa deve essere consentito a un solo processo per volta per evitare effetti di sovrapposizioni di esecuzione; se un processo accede al flag di una risorsa per vedere se è libera o per rilasciarla non può essere concesso ad un altro processo di esaminare il flag prima che sia stato completato l'aggiornamento.

Si supponga altrimenti che il flag segnali libero e che un processo possa impadronirsi della risorsa; se prima di poter aggiornare il flag un secondo processo lo potesse esaminare troverebbe la risorsa ancora libera ed entrambi i processi cercherebbero di impadronirsi della stessa risorsa contemporaneamente.

Le istruzioni di accesso al semaforo devono perciò formare una sequenza indivisibile, cioè non devono accettare interruzioni durante l'esecuzione di queste istruzioni (ciò viene realizzato disabilitando gli interrupt, cioè mascherando la CPU).

Queste sequenze di istruzioni vengono dette regioni critiche.

STALLO

La presenza di più programmi che competono per l'uso delle risorse può causare situazioni di stallo, cioè situazioni in cui non è possibile far avanzare nessun processo.

In particolare ciò avviene quando c'è una situazione di attesa circolare di risorse, cioè quando esiste una catena di processi in cui ogni processo richiede una risorsa occupata dal processo successivo nella catena. Per verificare l'esistenza di una situazione di attesa circolare si può costruire il grafo di allocazione delle risorse in cui i nodi rappresentano le risorse e i processi e gli archi collegano i processi alle risorse con un'orientazione diversa per indicare il possesso e la richiesta.

Il grafo di allocazione varia durante l'evoluzione dei processi; se si formano cicli significa che i processi del ciclo si trovano in condizione di stallo.

Se si verifica uno stallo bisogna far terminare forzatamente uno dei processi coinvolti in modo che gli altri possano avanzare; la terminazione può essere forzata manualmente dall'operatore, o nei sistemi più sofisticati dallo stesso sistema operativo. Per evitare il dello stallo il sistema operativo può seguire delle strategie di allocazione delle risorse che impediscano il verificarsi di condizioni di attesa circolare.

La tecnica di preallocazione delle risorse richiede che tutte le risorse necessarie al processo vengano associate al momento del caricamento in memoria, con la possibilità di rilasciarle poi una alla volta quando non servono più; bisogna però determinare all'inizio tutte le risorse necessarie e inoltre le risorse possono risultare poco sfruttate.

La tecnica di prerilascio delle risorse impone invece che quando a un processo viene rifiutata una risorsa, il processo rilasci anche tutte le altre risorse già associate. Questo metodo non si può comunque applicare per l'uso di risorse che non possono essere rilasciate come le stampanti (risorse non preemptive).

La soluzione più diffusa è quella di allocazione gerarchica delle risorse in cui tra le risorse viene definito un ordine gerarchico ed un processo può richiedere le risorse di cui necessita solo rispettando queste priorità. Anche in questo caso però bisogna conoscere tutte le risorse necessarie e ordinarle in modo corretto e si rischia comunque di usarle in modo poco efficiente.

Un'altra tecnica, nota come l'algoritmo del banchiere, consente di evitare lo stallo dando la possibilità al sistema operativo di rifiutare una risorsa a un processo se non restano abbastanza risorse disponibili per soddisfare la richiesta massima di risorse di almeno uno dei processi già attivi; perché ciò sia effettuabile deve essere noto in anticipo il numero massimo di risorse di cui ha bisogno ciascun processo.

In questo modo si assicura che almeno un processo possa terminare e rilasciare le risorse già occupate.