Sistemi Operativi

La gestione delle risorse

Introduzione

Il sistema operativo ha il compito di fornire la gestione dell'hardware ai programmi dell'utente.

Bootstrap

All’accensione la CPU esegue la procedura di caricamento in memoria del kernel, la parte del sistema operativo che contiene i moduli per la gestione dei dispositivi fisici: memoria, disco, elementi di I/O, ecc.

Quando il sistema operativo è stato caricato in memoria, il programma di boot lo manda in esecuzione

A questo punto il sistema operativo ha il controllo della macchina.

Esecuzione di un programma

Il sistema operativo che riceve la richiesta di esecuzione di un programma cede temporaneamente il controllo della CPU al programma utente e lo riprende al termine del programma.

Il programma utente richiede un servizio al sistema operativo mediante una chiamata di sistema.

Protezione

La CPU possiede due modalità di esecuzione:

- Modalità utente, la CPU esegue le istruzioni del programma
- Modalità di sistema, la CPU esegue istruzioni privilegiate, ovvero quelle che richiedono modifiche o accesso a informazioni del sistema (tabelle di interruzione, mappa di I/O, ecc.)

Protezione

Il programma utente non ha accesso diretto all'hardware del computer, ma deve chiedere al sistema operativo di comunicare con il dispositivo hardware.

Il sistema operativo riconosce la legittimità della richiesta e, modificando la modalità di funzionamento della CPU, svolge il servizio richiesto.

Programma

Un **programma** esprime l'algoritmo risolutivo di un problema, cioè contiene l’elenco dei calcoli da eseguire sui dati iniziali noti per determinare un risultato, stabilisce quindi una relazione tra i dati di ingresso e i dati di uscita.

Tra le varie notazioni simboliche usate per rappresentare un programma le più diffuse sono i diagrammi di flusso e lo pseudolinguaggio. Queste servono al programmatore per pervenire alla codifica del programma, non possono rappresentare lo *stato* del sistema di elaborazione associato al programma in esecuzione.

Processore

Il **processore**, anche se contiene più unità di controllo, esegue sequenzialmente le istruzioni di un programma. Esso cioè, a partire da una certa configurazione delle variabili in memoria e nei suoi registri interni, per ogni istruzione che esegue produce una modifica delle variabili e dei dispositivi coinvolti nell’operazione.

Processo

Il **processo**, associato ad un programma, è la descrizione delle *operazioni* svolte dalla CPU per eseguire le istruzioni del programma.

Il processo rappresenta quindi l'esecuzione delle istruzioni di un programma da parte della CPU.

Un processo è associato ad un programma ed è in esecuzione su una CPU.

Un processo è identificato da un PID (Process Identifier) e possiede una priorità.

Risorsa

L’esecuzione delle istruzioni di un programma comporta l’utilizzo di componenti e di bus del calcolatore.

Un componente che condiziona l’avanzamento di un processo è detto risorsa.

Esecuzione sequenziale

Esecuzione sequenziale di un programma: le operazioni vengono elencate una dopo l'altra e la CPU, prelevandole dalla memoria, le esegue nell'ordine in cui le trova. Un'operazione inizia solo quando la precedente è stata completata.

In un sistema di elaborazione la CPU è affiancata da alcuni dispositivi progettati per l'esecuzione di compiti specifici, dai quali ottiene collaborazione in casi particolari, per esempio:

- il coprocessore matematico, svolge in hardware gli algoritmi di calcolo delle funzioni matematiche, altrimenti svolti da sottoprogrammi,
- la controller del disco, provvede a indirizzare un settore e a trasferire i dati in un modo più veloce di quanto possa fare la CPU,
- il componente DMA, esegue il trasferimento di un elenco di dati, da una sorgente a una destinazione, in forma diretta, laddove la CPU introdurrebbe un trasferimento intermedio in un suo registro interno.

La CPU può continuare ad eseguire le istruzioni dei programmi, mentre questi processori specializzati possono eseguono le loro operazioni .

Multitasking

Il processore dispone di un timer che inizializza con un numero prestabilito e decrementa con una certa frequenza. Quando il timer raggiunge il valore 0 genera un segnale di interrupt.

Il sistema operativo effettua una **commutazione di processo** e inizializza nuovamente il timer.

La durata dell'intervallo è detta **quanto**.

Il sistema operativo assegna l'uso di un processore ad un processo per un quanto di tempo.

Se ci sono M processi ed N processori, con M>N, in un certo istante solo N processi sono in esecuzione, ma in un intervallo di tempo, formato da più quanti di tempo, sono stati eseguiti M processi, dando l'impressione che sono stati eseguiti contemporaneamente.

Commutazione di processo

Il processo che perde l’uso del processore è detto processo uscente.

Il processo che ha il turno di usare il processore è detto processo entrante.

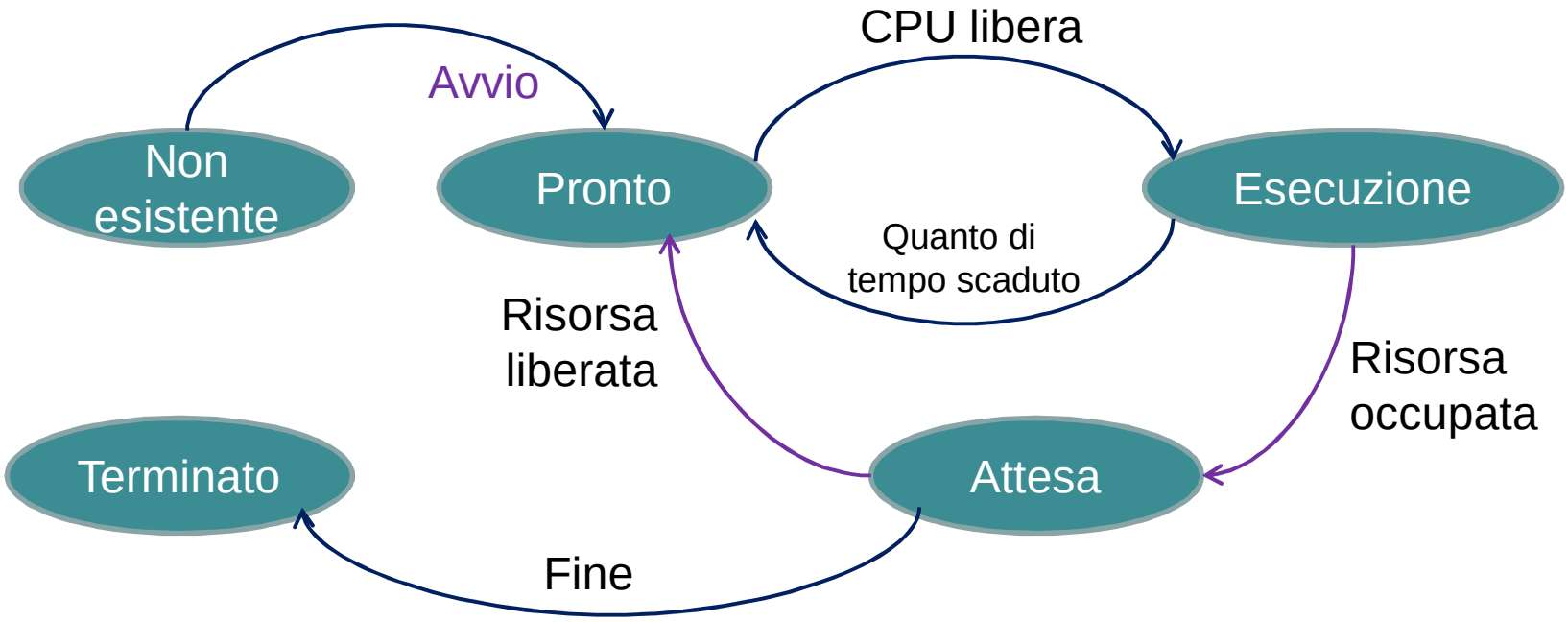
Quando un processo perde l’uso del processore, il sistema operativo salva, in un’area di memoria riservata, il contesto del processo uscente, ovvero l’immagine dei registri della CPU e, prima di assegnare la CPU al processo entrante, copia nei registri della CPU l’immagine dei registri del processo entrante, che erano stati salvati al momento dell’uscita.

Stati di un processo.

- Non iniziato o terminato
- Esecuzione: il processo è in esecuzione su un processore
- Pronto: il processo possiede tutte le risorse necessarie, deve avere solo l’uso del processore
- Attesa: il processo non può andare in esecuzione perché ha bisogno di una risorsa che è risultata assegnata ad un altro processo.

Stati di un processo

Un processo viene avviato, esegue le sue operazioni e poi termina. Durante l'esecuzione il processo potrebbe trovarsi nello stato di attesa e nello stato di pronto. Le transizioni da unmo stato all'altro sono determinate dal verificarsi di alcuni eventi



Transizioni di stato

Un processo passa dallo stato di esecuzione alla stato pronto quando scade il quanto di tempo assegnato.

Un processo passa dallo stato di esecuzione allo stato di attesa quando:

- Ha bisogno di una risorsa utilizzata da un altro processo,
- Chiede un'operazione di I/O.

Il sistema operativo sposta un processo dallo stato di attesa allo stato pronto quando la risorsa che aveva chiesto si è liberata.

Politica di scheduling

- Il Sistema operativo gestisce due liste di processi: una lista di attesa e una lista dei processi pronti.
- La **politica di scheduling** è il criterio secondo cui il sistema operativo seleziona, dalla lista dei processi *pronti*, un processo da mandare in esecuzione

Context switch

- Il **contesto** di un processo è rappresentato dai valori immagazzinati nei registri interni della CPU, e il sistema operativo riserva, ad ogni processo attivo, un segmento privato in cui memorizzare il contesto.
- Gli indirizzi base di tali segmenti, sparsi per la memoria, sono riepilogati in una tabella. L'indice che permette di accedere all'indirizzo base di un segmento di stato è usato dal sistema operativo per identificare il processo.
- Il *task register*, nella CPU, viene aggiornato dal sistema operativo con l'identificatore del processo che passa in esecuzione, e la CPU automaticamente recupera il contesto del processo dal segmento di stato.

Algoritmi di sheduling

Lo scheduler è il componente del sistema operativo che ha il compito di prelevare dalla coda dei processi pronti il descrittore del processo da mandare in esecuzione.

La coda dei processi viene gestita con il criterio FIFO (First In First Out), ovvero rispettando l’ordine di arrivo dei processi ed assegnando a rotazione l’uso della CPU

Livelli di scheduling

- Long term scheduling: determina quali programmi devono essere caricati dalla memoria di massa alla memoria principale.
- Medium term scheduling: rimuove temporaneamente dalla memoria principale un processo in esecuzione.
- Short term scheduling: seleziona tra i processi pronti in memoria principale quello a cui assegnare la CPU.

Algoritmi di short term scheduling

- Round Robin: La coda dei processi è gestita il criterio FIFO, ma ad ogni processo è assegnata la CPU per un quanto di tempo q prefissato. q deve essere sufficientemente grande rispetto al tempo di cambio del contesto.
- Priorità statica: fissata alla creazione dei processi in base alle loro caratteristiche:
- processi foreground (sistemi interattivi) ---> alta priorità.
- processi background (batch) ---> bassa priorità.
- Priorità dinamica: può essere modificata durante l’esecuzione dei processi

Algoritmo Shortest Job First

Per alcune operazioni, il sistema operativo è in grado di stabilire il tempo di esecuzione richiesto. L’algoritmo SJF preleva i processi dalla coda privilegiando il più breve.

- occorre conoscere le caratteristiche dei processi.
- minimizza il tempo medio di permanenza di un processo nel sistema (dall’avvio al termine).