

Gestione dei processi

❖ II) Scheduling

Scheduling dei processi

□ Scheduling

❖ Tecnica di allocazione delle risorse

- Il S.O. alloca le risorse tra le potenziali necessità concorrenti di molti processi
- Lo scheduling è una funzione fondamentale dei S.O.
 - Si sottopongono a scheduling quasi tutte le risorse di un calcolatore

□ Scheduler della CPU

❖ Componente più importante del kernel

❖ Componente software che gestisce l'avvicendamento dei processi

- Lo **scheduling della CPU** consiste nella scelta di un processo cui assegnare la CPU: l'effettiva assegnazione della CPU al processo prescelto è effettuata dal **dispatcher**
- Lo scheduling della CPU è alla base dei sistemi multiprogrammati: più processi sono mantenuti in memoria e la CPU è assegnata loro dinamicamente

Scheduling dei processi

□ Classi di scheduler

1. Scheduler a breve termine

- Sceglie tra i **processi pronti** quello a cui assegnare la CPU
- Interviene quando il processo in esecuzione perde il controllo della CPU

2. Scheduler a medio termine (Swapping)

- Gestisce i **processi bloccati** per lunghe attese
- Trasferisce temporaneamente in memoria secondaria l'immagine di tali processi al fine di ottimizzare l'uso della memoria principale

3. Scheduler a lungo termine

- Sceglie nella memoria secondaria quali programmi caricare in memoria centrale in modo da creare i processi corrispondenti
- È una componente importante dei sistemi batch multiprogrammati

□ Dispatcher

- ❖ Modulo del S.O. che passa effettivamente il controllo della CPU ai processi scelti dallo scheduler a breve termine

- Effettua il cambio di contesto di elaborazione, il passaggio al modo utente e il salto alla giusta posizione del programma utente per riavviare la esecuzione
- Il tempo richiesto dal dispatcher per fermare un processo ed avviare la esecuzione di un altro processo definisce la **latenza del dispatcher**

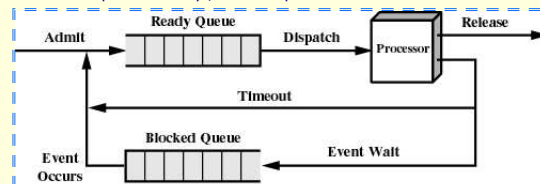
Scheduling dei processi

□ Scheduling nei sistemi multiprogrammati a singolo processore

- ❖ Tutte le volte che un processo entra nel sistema viene posto in una delle **code** gestite dallo scheduler

1. Code di accodamento dei processi in uno stato

- ❖ Coda dei processi ready, Coda dei processi blocked



- ❖ Gli **elementi delle code** sono i descrittori di processo (**PCB**)

- Coda FIFO
- Coda con priorità
- Coda a lista concatenata
 - Una intestazione della coda dei processi pronti contiene i puntatori al primo e all'ultimo PCB dell'elenco, e ciascun PCB è esteso con un campo puntatore che indica il successivo processo contenuto nella lista dei processi pronti

Scheduling dei processi

2. Selezione dei processi dalle code

- Livelli di priorità
- Algoritmi di scheduling

❖ Livelli di priorità dei processi

- Priorità legata al tipo di utente che detiene il processo
- **Livello utente:** processi responsabili dell'esecuzione di programmi utente
- **Livello supervisore:** processi responsabili di alcune funzioni del SO
- **Livello di I/O:** processi di servizio degli interrupt
- **Livello delle eccezioni:** processi responsabili della integrità del sistema e processi che gestiscono errori ed eccezioni dovute ad esecuzione non corretta di programmi utente



Scheduling dei processi

□ Algoritmi di scheduling

- ❖ Algoritmi di selezione del processo a cui assegnare la CPU, per quanto tempo e in quale momento
- ❖ Criteri di scheduling
 - **Utilizzo della CPU**
 - La CPU deve essere utilizzata in modo da garantire che ogni processo abbia un'equa porzione di tempo di CPU
 - **Produttività**
 - Deve risultare massimo il **throughput** (numero di lavori eseguiti nell'unità di tempo)
 - Criterio non importante per sistemi a singolo utente
 - **Tempo di completamento**
 - L'intervallo di tempo tra la sottomissione del processo e il completamento (**turnaround time**) deve essere il più breve possibile
 - Deve essere minimizzata la somma degli intervalli di tempo passati nella coda dei processi pronti
 - **Tempo di risposta**
 - L'intervallo di tempo tra la sottomissione di una richiesta e la prima risposta prodotta deve essere il più breve possibile
 - Criterio importante nei sistemi interattivi

Scheduling dei processi

□ Algoritmi di scheduling

- ❖ Diversi algoritmi di scheduling che hanno proprietà differenti e possono favorire una particolare classe di processi

1. **Scheduling in ordine di arrivo**
 - First Come First Served
2. **Scheduling circolare**
 - Round Robin
3. **Scheduling per brevità**
 - Shortest Process Next
4. **Scheduling per priorità**
 - Priority
5. **Scheduling a code multiple**
 - Code multiple
6. **Scheduling a code multiple con reazione**

Scheduling dei processi

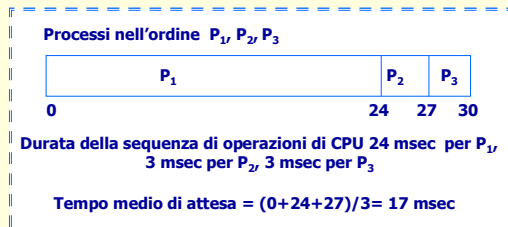
□ Algoritmo "First Come First Served"

❖ Scheduling in ordine d'arrivo

- ❖ Tale algoritmo assegna la CPU al processo che la richiede per primo
- ❖ La realizzazione del criterio FCFS si fonda su una **codice FIFO**
 - Quando un processo è inserito nella coda ready si collega il suo PCB all'ultimo elemento della coda
 - Quando la CPU è libera, è assegnata al processo che si trova alla testa della coda ready, rimuovendolo da essa
- ❖ FCFS è senza prelazione
 - Una volta che la CPU è stata assegnata ad un processo, questo la trattiene fino al momento del rilascio, che può essere al termine dell'esecuzione o alla richiesta di un'operazione di I/O
- ❖ FCFS tende a favorire i processi processor-bound rispetto a quelli I/O-bound
 - Non adatto ai sistemi a partizione di tempo

Scheduling dei processi

- ❖ Il tempo medio di attesa di FCFS è abbastanza lungo



Scheduling dei processi

□ Algoritmo "Shortest Process Next"

❖ Scheduling per brevità

- ❖ Tale algoritmo associa ad ogni processo la **lunghezza della successiva sequenza di operazioni** della CPU

- Necessità di determinare per ogni processo la lunghezza della successiva sequenza di operazioni della CPU e non la lunghezza totale
- La CPU è assegnata al processo che ha la più breve lunghezza della successiva sequenza di operazioni della CPU
- Se due processi hanno le successive sequenze di operazioni della CPU della stessa lunghezza, si applica lo scheduling FCFS

- ❖ SPN può essere senza prelazione o con prelazione

- **SPN con prelazione (Shortest Remaining Time First)** sostituisce il processo in esecuzione con un nuovo processo della coda ready, con sequenza di elaborazione più breve rispetto a quella del processo in esecuzione

Scheduling dei processi

- ❖ SPN rende minimo il **tempo di attesa medio** per un insieme di processi
 - Spostando un processo breve prima di un processo lungo, il tempo di attesa di un processo breve diminuisce più di quanto aumenti il tempo di attesa per il processo lungo

Processi ordinati dall' algoritmo P_4, P_1, P_3, P_2

P_4	P_1	P_3	P_2	
0	3	9	16	24

Durata della sequenza di operazioni di CPU: 6 msec per P_1 ,
8 msec per P_2 , 7 msec per P_3 , 3 msec per P_4

Tempo medio di attesa = $(3+16+9+0)/4 = 7$ msec

Scheduling dei processi

- ❖ SPN deve conoscere la lunghezza della successiva sequenza di istruzioni
 - Difficoltà nella determinazione a priori della lunghezza della successiva sequenza di istruzioni
 - Non applicabile allo scheduling a breve termine
 - Difficile applicabilità a processi interattivi di durata non nota a priori
- ❖ Approssimazione dello scheduling SPN mediante **predizione** della lunghezza della successiva sequenza di istruzioni
 - Il valore approssimato della lunghezza è determinato calcolando la media esponenziale delle effettive lunghezze delle precedenti sequenze di operazioni della CPU

t_n lunghezza della n-esima sequenza
 T_{n+1} valore previsto della lunghezza della n-esima sequenza
 $0 \leq \alpha \leq 1$

$$T_{n+1} = \alpha t_n + (1 - \alpha) T_n$$

Scheduling dei processi

□ **Algoritmo "Priority"**

❖ **Scheduling per priorità**

- ❖ Tale algoritmo assegna la CPU al processo che ha la **priorità più alta**
 - Ad ogni processo viene assegnata una priorità
 - I processi con uguale priorità sono schedulati con l'algoritmo FCFS
- ❖ Le priorità possono essere definite
 - **Internamente dal S.O.**, in base a una o più quantità misurabili
 - Limiti di tempo, requisiti di memoria, numero dei file aperti, tipo di processo (I/O bound o CPU bound)
 - **Esternamente dall'utente**, secondo criteri esterni al S.O.
 - Importanza del processo...
- ❖ L' algoritmo Shortest Remaining Time First è un algoritmo Priority in cui la priorità è l' inverso della lunghezza prevista della successiva sequenza di operazioni

Scheduling dei processi

- ❖ Lo scheduling per priorità può essere senza prelazione o con prelazione
 - **Scheduling per priorità con prelazione**
 - La CPU è sottratta al processo attualmente in esecuzione se il nuovo processo della coda ready ha una priorità più alta
 - Si può verificare attesa indefinita per processi con priorità bassa se vi è un flusso di processi con priorità maggiore
 - **Scheduling per priorità senza prelazione**
 - La CPU non è sottratta al processo attualmente in esecuzione anche se il nuovo processo della coda ready ha una priorità più alta: quest'ultimo è posto alla testa della coda dei processi pronti

Processi arrivati nell'ordine P₁, P₂, P₃, P₄, P₅

P ₂	P ₅	P ₁	P ₃	P ₄	
0	1	6	16	18	19

Durata della sequenza di operazioni di CPU: 10 msec per P₁ con priorità 3,
1 msec per P₂ con priorità 1, 2 msec per P₃ con priorità 4, 1 msec per P₄
con priorità 5, 5 msec per P₅ con priorità 2

Tempo medio di attesa = 8 msec

Scheduling dei processi

□ **Algoritmo "Round Robin"**

❖ **Scheduling circolare**

- ❖ Tale algoritmo assegna la CPU ad ogni processo per un intervallo di tempo costante, detto **quanto di tempo** (time slice)
 - Tipicamente 10-100 msec
 - Non appena termina il quanto di un processo, esso è posto in stato di ready alla fine della coda dei processi pronti e la CPU è assegnata ad un altro processo
- ❖ Lo scheduling **RR** gestisce la lista dei processi in stato di ready come una **coda circolare**
- ❖ RR è con **prelazione**
 - I processi I/O bound sono penalizzati rispetto a quelli CPU bound
- ❖ Le prestazioni dell'algoritmo RR dipendono molto dalla dimensione del quanto di tempo
 - Se il quanto di tempo è molto lungo, il criterio di scheduling RR si riduce al criterio FCFS
 - Se il quanto di tempo è molto breve, il carico di scheduling dovuto al tempo dei cambi di contesto diventa eccessivo

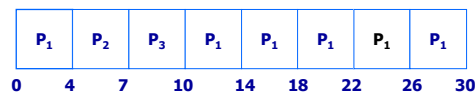
Scheduling dei processi

- ❖ Se il quanto di tempo è molto breve, il criterio di scheduling RR si chiama **condivisione della CPU**
 - Gli utenti hanno l'impressione che ciascuno degli n processi disponga di una CPU a 1/n della velocità della CPU reale
 - La CPU sarà impegnata per gran parte del tempo nel controllo dei frequenti passaggi da un processo ad un altro
- ❖ Il tempo di attesa per il criterio di scheduling RR è abbastanza lungo

Insieme dei processi

P₁ con durata della sequenza di operazioni di CPU di 24 msec
P₂ con durata della sequenza di operazioni di CPU di 3 msec
P₃ con durata della sequenza di operazioni di CPU di 3 msec

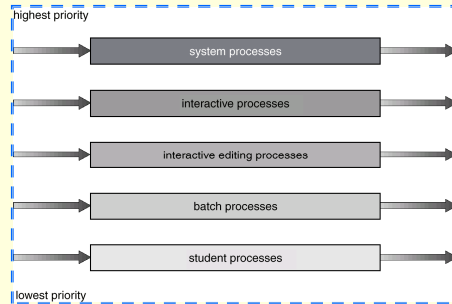
Quanto di tempo di 4 msec



Scheduling dei processi

□ Algoritmo "Code multiple"

- ❖ La coda dei processi pronti è suddivisa in **code distinte**



- ❖ I processi sono assegnati ad una coda secondo alcune caratteristiche del processo come la quantità di memoria richiesta, la priorità o il tipo di processo

Scheduling dei processi

- ❖ Distinzione tra code di processi

1. Coda dei **processi in primo piano**, o interattivi (**foreground process**)
2. Coda dei **processi in secondo piano (background process)** o a lotti, o batch
 - Ogni coda ha il suo algoritmo di scheduling
 - Coda dei processi interattivi con scheduling RR
 - Coda dei processi batch con scheduling FCFS

- ❖ Lo **scheduling fra le code** è normalmente con priorità fissa e con prelazione

- Ogni coda ha priorità assoluta sulle code di priorità più bassa
- Ad ogni coda possono essere assegnati differenti quanti di tempo

- ❖ Nello scheduling a code multiple i processi sono assegnati **permanentemente** ad una coda e non possono passare da una coda ad un'altra

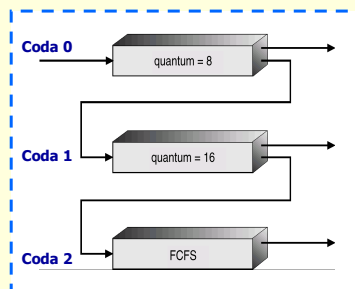
- Impostazione rigida con il vantaggio di un basso carico di scheduling

Scheduling dei processi

□ **Algoritmo a code multiple con retroazione**

- ❖ Tale algoritmo consente ai processi lo **spostamento dinamico fra le code**
 - Differenti metodi per determinare quando spostare un processo in una coda con priorità più elevata o quando spostare un processo in una coda con priorità più bassa
- ❖ Ai processi di code con priorità più elevata è assegnato un numero di quanti inferiore: non appena un processo termina tutti i suoi quanti, viene spostato nella classe con priorità inferiore (n.ro di quanti maggiore)
 - Per ogni processo si riduce il numero di passaggi allo stato di ready per concludere la sua esecuzione
 - Man mano che un processo scende nelle classi, otterrà la CPU sempre meno frequentemente lasciandola libera per i processi che richiedono poco tempo di CPU
- ❖ Il criterio di scheduling che si avvale delle code multiple con reazione è il più generale ma anche il più complesso

Scheduling dei processi



- Lo scheduler fa eseguire tutti i processi nelle coda 0. I processi che non terminano entro il quanto di tempo vengono spostati alla fine della coda 1
- Quando la coda 0 è vuota si eseguono i processi della coda 1. Se le code 0 e 1 sono vuote si eseguono i processi della coda 2
- Un processo in ingresso alla coda 0 ha prelazione sulla coda 1. Un processo in ingresso alla coda 1 ha prelazione sulla coda 2

Tale scheduling dà la priorità massima ai processi con una sequenza di operazioni di CPU di non più di 8 msec

Scheduling dei processi

- ❖ Uno scheduler a code multiple con retroazione è caratterizzato da diversi parametri
 1. Numero di code
 2. Algoritmo di scheduling per ciascuna coda
 3. Metodo usato per determinare quando spostare un processo in una coda con priorità maggiore
 4. Metodo usato per determinare quando spostare un processo in una coda con priorità minore
 5. Metodo usato per determinare in quale coda si deve spostare un processo quando richiede un servizio

Scheduling dei processi

- **Scheduling a due livelli**
 - ❖ Solo un sottoinsieme di processi è tenuto in memoria, i rimanenti sono tenuti temporaneamente su disco
 - ❖ Due livelli di schedulazione
 - **Scheduling ad alto livello**
 - Sono rimossi dalla memoria i processi che vi sono stati per un tempo sufficientemente elevato per caricare un nuovo insieme di processi dal disco (**swapping**)
 - **Scheduling a basso livello**
 - Si applica lo scheduling all'insieme dei processi attualmente presenti in memoria