

- Si assuma che un sistema di schedulazione sia formato da due code. La prima coda è gestita con Round Robin con un quanto di dimensione 1 e contiene i processi che terminano entro 4 unità temporali. Se un processo dura più di 4 unità temporali viene messo nella seconda coda gestita con strategia FIFO.

Si calcoli il tempo di turnaround medio e si visualizzi la sequenza di schedulazione se arrivano i seguenti processi:

Processo	Istante di arrivo	Durata
P1	1	3
P2	0	5
P3	3	4

- Si consideri il seguente insieme di 4 processi:

Processo	Arrivo	Durata
P1	0	4
P2	1	2
P3	1	2
P3	4	2

Assumendo che non si sia overhead di ContextSwitch, calcolare il tempo di turnaround medio per le politiche di schedulazione FIFO, SJF (Shortest Job First), SRTF (Shortest Remaining Time First), RR (Round Robin) con quanto pari a 1.

(Nel caso si debba scegliere tra due processi equivalenti, si scelga quello con numero minore)

- Da un monitoraggio sulla esecuzione di un processo risulta la seguente sequenza di CPU bursts, cioè tempi di esecuzione prima di una richiesta di I/O: $t_0=50\text{ms}$, $t_1=45\text{ms}$, $t_2=43\text{ms}$. Qual'è il tempo di esecuzione predetto al prossimo istante usando la media esponenziale con $\alpha=0.5$?

(nota: la media esponenziale è descritta dalla: $\tau_{n+1} = \alpha t_n + (1 - \alpha) \tau_n$ dove t è il tempo di esecuzione misurato e τ è il tempo predetto. Si ricordi che $\tau_0=0$)

- Descrivere la differenza tra schedulazione preemptive e schedulazione non-preemptive, enunciando almeno un vantaggio/uno svantaggio per ciascuna delle due.

Un algoritmo di schedulazione di n processi stabilisce l'ordine con il quale i processi devono essere eseguiti. Se un algoritmo di schedulazione nonpreemptive deve eseguire n processi su una sola CPU, qual'è il numero totale delle possibili schedulazioni?

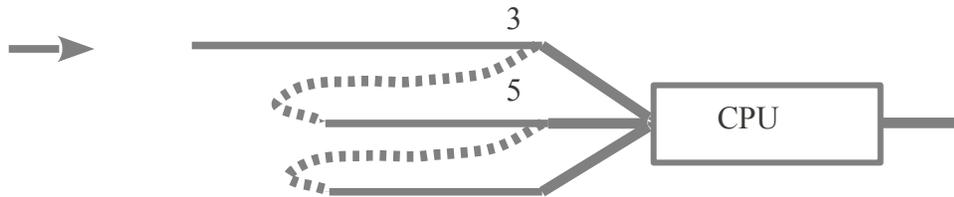
- Supponiamo di avere tre processi, i cui istanti di arrivo e durate sono descritti dalla seguente tabella:

Processo	arrivo	durata
P1	0.0	6
P2	1.0	4
P3	2.5	1

Rispondere alle seguenti due domande:

- Disegnare i diagrammi temporali di schedulazione con gli algoritmi FIFO, SJF (shortest job first), SRTF (shortest remaining time first) e RR (round robin) con Quanto = 1.
- Calcolare il tempo medio di attesa in coda se i tre processi sono schedulati con FIFO, SJF, SRTF, RR.

- Gli stessi tre processi del punto precedente sono schedulati con un algoritmo Multilevel Feedback Queue semplificato, che usa tre code gestite con modalità FIFO.



L'algoritmo funziona nel seguente modo: tutti i processi arrivano nella prima coda; se un processo che parte nella prima coda esegue (senza interruzioni) per meno di 3 unità temporali, esce direttamente dalla 1° coda, altrimenti viene spostato nella 2° coda. Se un processo che risiede nella 2° coda termina in meno di 5 unità temporali, esce dalla 2° coda, altrimenti viene messo nella 3° coda, dove eseguono solo se non ci sono altri processi nelle code precedenti e vengono interrotti da arrivi nelle code precedenti.

Ciò premesso, disegnare il diagramma temporale di schedulazione (diagramma di Gantt) e calcolare il tempo di turnaround medio nell'ipotesi di trascurare il tempo di context switch.

- In un certo istante t , diciamo $t=0$, nella coda dei processi pronti sono presenti i seguenti descrittori di processo, nell'ordine: P1 P2 P3 P4 P5 (P1 è verso la CPU)

Ogni descrittore contiene i seguenti dati:

– Cpu burst (sec) effettivi dell'ultima esecuzione: 10 – 6 – 2 – 4 – 8

– Cpu burst predetti dell'ultima esecuzione: 8 - 4 - 1 - 5 - 9

Usare la media esponenziale (con $w=0.5$) per predire il prossimo Cpu burst e calcolare il tempo medio di attesa e il tempo medio di turnaround con la politica SJF.

Quale sarebbero i due tempi se la schedulazione fosse realizzata con la politica FIFO?

- Calcolare il tempo medio di attesa per il seguente insieme di processi utilizzando gli algoritmi SRJF (Shortest Remaining Job First) e Round Robin con $quantum=4$, valutando anche il numero di context switch.

Processo	arrivo	CPU burst
P1	0	8
P2	2	2
P3	2	4
P4	4	2
P5	5	3

Si abbiano i seguenti thread da schedulare in tempo reale, cioè entro le loro deadline:

arrivo durata deadline

T1 0 1 3

T2 1 2 4

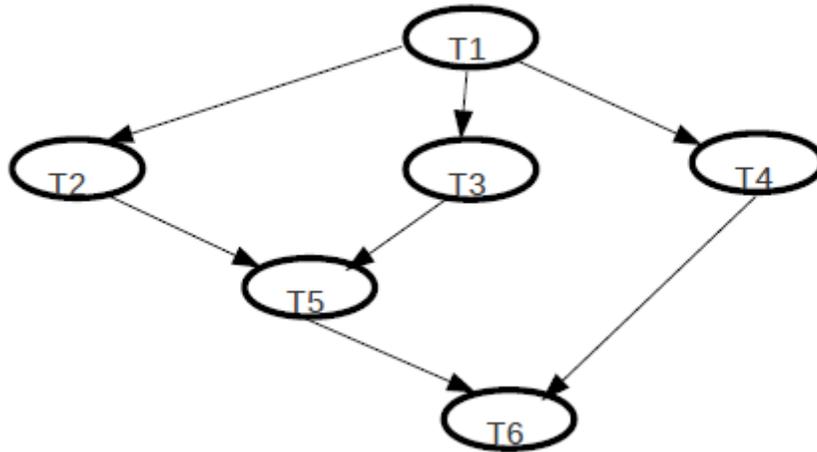
T3 1 1 6

T4 2 2 7

T5 3 1 6

T6 3 1 8

Inoltre, i thread siano sottoposti ai seguenti vincoli di precedenza:



Schedulare i thread con l'algoritmo non pre-emptive EDF calcolando la massima lateness.

3. 4pt. Si abbiano i seguenti thread da schedulare in tempo reale:

arrivo durata deadline

T1 0 1 3

T2 2 2 5

T3 1 1 6

T4 3 2 9

T5 2 1 8

Schedulare i thread usando l'algoritmo Earliest Deadline First preemptive.