

Sistemas Operativos

Práctica 2: *Scheduling*

Notas preliminares

- Los ejercicios marcados con el símbolo ★ constituyen un subconjunto mínimo de ejercitación. Sin embargo, aconsejamos fuertemente hacer todos los ejercicios.

Parte 1 – Políticas clásicas de *scheduling*

Ejercicio 1

La siguiente secuencia describe la forma en la que un proceso utiliza el procesador.

Tiempo	Evento
0	<i>load store</i>
1	<i>add store</i>
2	<i>read</i> de archivo
3	espera E/S
..	..
10	espera E/S
11	<i>store increment</i>
12	inc
13	<i>write</i> en archivo
14	espera E/S
..	...
20	espera E/S
21	<i>load store</i>
22	<i>add store</i>

- Identificar las ráfagas de CPU y las ráfagas de E/S.
- ¿Qué duración tiene cada ráfaga?

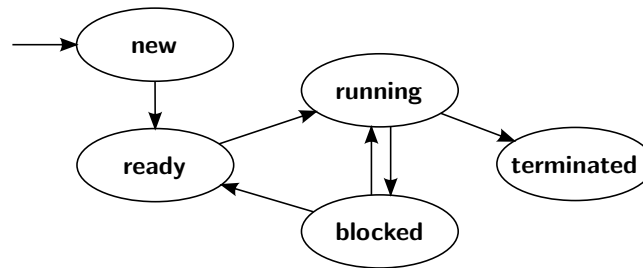
Ejercicio 2 ★

Sean P_0 , P_1 y P_2 tales que

- P_0 tiene ráfagas cortas de E/S a ciertos dispositivos.
 - P_1 frecuentemente se bloquea leyendo de la red.
 - P_2 tiene ráfagas prolongadas de alto consumo de CPU y luego de escritura a disco.
- Para planificar estos procesos, ¿convendría usar un algoritmo de Round Robin? ¿convendría usar uno de prioridades? Justifique su respuesta.

Ejercicio 3

¿A qué tipo de *scheduler* corresponde el siguiente diagrama de transición de estados de un proceso?

**Ejercicio 4 ★**

¿Cuáles de los siguientes algoritmos de *scheduling* pueden resultar en *starvation* (inanición) y en qué condiciones?

- Round-robin*.
- Por prioridad.
- SJF.
- SRTF.
- FIFO.
- Colas de multinivel.
- Colas de multinivel con feedback (aging).

Ejercicio 5

Considere una modificación a *round-robin* en la que un mismo proceso puede estar encolado varias veces en la lista de procesos *ready*. Por ejemplo, en un RR normal se tendrían en la cola *ready* a P1,P2,P3,P4, con esta modificación se podría tener P1,P1,P2,P1,P3,P1,P4.

- ¿Qué impacto tendría esta modificación?
- Dar ventajas y desventajas de este esquema. Piense en el efecto logrado, no en la forma de implementarlo.
- ¿Se le ocurre alguna otra modificación para mantener las ventajas sin tener que duplicar las entradas en la lista de procesos *ready*?

Ejercicio 6

Considerar el siguiente conjunto de procesos:

Proceso	Ráfaga de CPU	Prioridad
P ₁	10	3
P ₂	1	1
P ₃	2	3
P ₄	1	4
P ₅	5	2

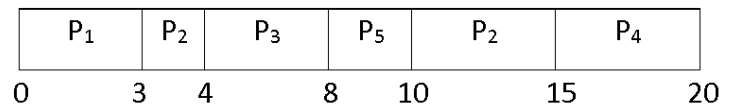
Se supone que los procesos llegan en el orden P₁, P₂, P₃, P₄, P₅ en el instante 0.

- Dibujar los diagramas de Gantt para ilustrar la ejecución de estos procesos usando los algoritmos de scheduling FCFS, SJF, con prioridades sin desalojo (a menor el número, mayor la prioridad), round-robin (*quantum* de 1 unidad de tiempo, ordenados por el número de proceso).
- ¿Cuál es el waiting time promedio y de turnaround promedio para cada algoritmo?
- ¿Cuál de los algoritmos obtiene el menor waiting time promedio, y el menor turnaround?

Ejercicio 7

El siguiente diagrama de Gantt corresponde a la ejecución tres procesos en un sistema monoprocesador.

Process	Burst Time(CPU)	Arrival Time(ms)
P ₁	3	0
P ₂	6	2
P ₃	4	4
P ₄	5	6
P ₅	2	8



- Calcular el *waiting time* y el *turnaround* promedios.
- Indicar de qué tipo de *scheduler* se trata, justificando claramente esa conclusión.

Ejercicio 8

Suponiendo que los siguientes procesos llegan en los tiempos indicados.

Proceso	Ráfaga de CPU	Instante de llegada
P ₁	8	0.0
P ₂	4	0.4
P ₃	1	1.0

- ¿Cuál es el tiempo de *turnaround*^a promedio para estos procesos usando FCFS?
- ¿Cuál es usando SJF?
- SJF se supone que mejora la performance, pero al elegir ejecutar P₁ inicialmente no había forma de saber que iban a llegar dos cortos luego. Volver a calcular el tiempo de *turnaround* promedio pero dejando el procesador *idle* por una unidad de tiempo y luego usar SJF.

^aRecordar que es el tiempo de finalización menos el de llegada.

Ejercicio 9

Considerar la siguiente tabla de procesos:

Proceso	Ráfaga de CPU	Instante de llegada	Prioridad
P ₁	50 ms	0 ms	4
P ₂	20 ms	20 ms	1
P ₃	100 ms	40 ms	3
P ₄	40 ms	60 ms	2

- Mostrar la planificación de estos procesos utilizando SRTF^a, de prioridad con desalojo y *round-robin* con *quantum* de 30 ms.
- ¿Cuál es el *waiting time* promedio para cada uno de los algoritmos?

^a*Shortest Remaining Time First*, es el *SRT* con desalojo y diferentes tiempos de llegada. Le da prioridad a los procesos cuyo tiempo de ejecución restante es menor.

Ejercicio 10 ★

Para los procesos presentados en la siguiente tabla, realizar un gráfico de Gantt para cada una de los algoritmos de scheduling indicados:

- FCFS.
- RR (*quantum*=10).
- SJF.

Proceso	Ráfaga de CPU	Instante de llegada
P ₁	1	5
P ₂	10	6
P ₃	1	7
P ₄	10	8

Calcular el *waiting time* y el *turnaround* promedios para cada una de los algoritmos.

Ejercicio 11

Considere los siguientes procesos:

Proceso	Ráfaga de CPU	Instante de llegada
P ₁	8	0
P ₂	8	5
P ₃	6	14
P ₄	5	15

- a) Realizar un diagrama de Gantt para un algoritmo de scheduling *round-robin* con un *quantum* de 5 unidades de tiempo.
- b) Realizar un diagrama de Gantt para un algoritmo tipo *shortest remaining time first*.
- c) Calcular el tiempo de *turnaround* promedio en ambos casos.
- d) A pesar de que uno de los dos casos tiene un tiempo de *turnaround* promedio mucho menor, explicar por qué en algunos contextos podría tener sentido utilizar la otra política. Para esto considere distintos tipos de procesos: real time, interactivos, batch, etc.

Ejercicio 12

Considere los siguientes procesos:

Proceso	Ráfaga de CPU	Instante de llegada	Cola asignada
P ₁	4	0	1
P ₂	3	0	1
P ₃	8	0	2
P ₄	5	10	2

- a) Realizar un diagrama de Gantt para un algoritmo de scheduling *Multilevel feedback queue* con dos colas: una cola 1 con *quantum* de 1 unidad de tiempo, y una cola 2 con *FCFS*. La cola 1 tiene más prioridad que la 2. Usa política con desalojo. Para cada proceso se indica qué cola se le asigna en el momento de su llegada.
- b) Calcular el tiempo de *turnaround* promedio y el *waiting time* promedio.

Ejercicio 13

Considere un algoritmo de scheduling que favorece a aquellos procesos que han usado la menor cantidad de tiempo de procesador en el pasado reciente. ¿Explique por qué favorecería a los procesos que realizan muchas E/S, pero a la vez no dejaría a los intensivos en CPU en *starvation*?

Ejercicio 14

Considere el siguiente proceso y los tiempos de ráfagas de CPU para un algoritmo SJF:

Proceso	Ráfaga de CPU real	Ráfaga de CPU estimada
P ₁		6
P ₁	5	7
P ₁	7	8
P ₁		X

- a) Calcule el valor de X (tiempo de ráfaga estimado), considerando uno y dos ráfagas anteriores. Asuma $\alpha = 1/2$