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	*load store*
1	*add store*
2	*read* de archivo
3	espera E/S
..	..
10	espera E/S
11	*store increment*
12	inc
13	*write* en archivo
14	espera E/S
..	...
20	espera E/S
21	*load store*
22	*add store*

a) Identificar las ráfagas de CPU y las ráfagas de E/S.

b) ¿Qué duración tiene cada ráfaga?

Ejercicio 2 ★

Sean P_0, P_1 y P_2 tales que

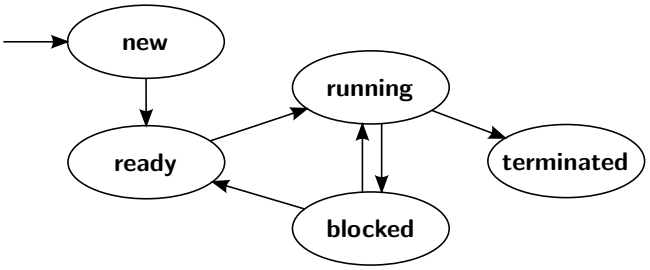
- P_0 tiene ráfagas cortas de E/S a ciertos dispositivos.
- P_1 frecuentemente se bloquea leyendo de la red.
- P_2 tiene ráfagas prolongadas de alto consumo de CPU y luego de escritura a disco.

a) Al planificar estos procesos, ¿cuánto *quantum* (mucho/poco) y qué prioridad (alta/baja) habría que asignarles y por qué?

b) Indicar a qué tipo de aplicación podría estar correspondiendo cada proceso.

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?

a) *Round-robin*.
b) Por prioridad.
c) SJF (trabajo más corto primero).
d) FIFO.

Ejercicio 5

Considerar una modificación a *round-robin* para que la lista de procesos *ready* sea una lista de punteros a las PCB de los procesos.

a) ¿Cuál es el efecto de poner dos punteros a la misma PCB?
b) Dar ventajas y desventajas del esquema del punto a). Piense en el efecto logrado, no en la forma de implementarlo.
c) ¿Se le ocurre alguna otra modificación para mantener las ventajas sin tener que duplicar punteros?

Ejercicio 6

Considerar el siguiente conjunto de procesos:

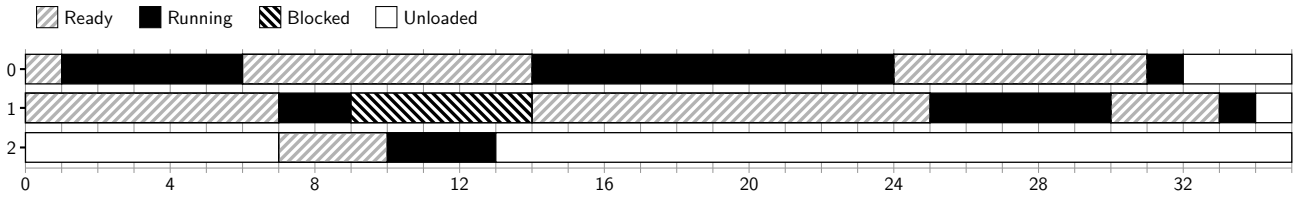
Proceso	Tiempo de ráfaga	Prioridad
P_1	10	3
P_2	1	1
P_3	2	3
P_4	1	4
P_5	5	2

Se supone que los procesos llegan en el orden P_1, P_2, P_3, P_4, P_5 en el instante 0.

a) Dibujar los diagramas de Gantt para ilustrar la ejecución de estos procesos usando los algoritmos de planificación FCFS, SJF, planificación por prioridades sin desalojo (un número de prioridad bajo indica una prioridad alta) y planificación por turnos (*quantum* de 1 unidad de tiempo).
b) ¿Cuál es el tiempo de espera y de ejecución de cada proceso para cada algoritmo de planificación?
c) ¿Cuál de los algoritmos obtiene el menor tiempo medio de espera?

Ejercicio 7

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



a) Calcular el *waiting time* y el *turnaround* promedios.
b) 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	Tiempo de procesamiento	Instante de llegada
P_1	8	0.0
P_2	4	0.4
P_3	1	1.0

a) ¿Cuál es el tiempo de *turnaround*[a] promedio para estos procesos usando FIFO?
b) ¿Cuál es usando SJF?
c) SJF se supone que mejora la perfomance, pero al elegir ejecutar P_1 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.

[a] Recordar que es el tiempo de finalización menos el de llegada.

Ejercicio 9

Considerar la siguiente tabla de procesos:

Proceso	Tiempo de procesamiento	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

a) Mostrar la planificación de estos procesos utilizando SRT[a], esquema de prioridad sin desalojo y *round-robin* con *quantum* de 30 ms.
b) ¿Cuál es el tiempo promedio de espera para cada uno de los algoritmos?

[a] *Shortest Remaining Time*. 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 las politicas de planificación indicadas:

- FCFS.

- RR (*quantum*=10), con un costo de cambio de contexto de 1 unidad de tiempo.
- SJF.

Proceso	Tiempo de procesamiento	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 las políticas.

Ejercicio 11

Considerar los siguientes procesos:

Proceso	Tiempo de ejecución total	Instante de llegada
P_1	8	0
P_2	8	5
P_3	6	14
P_4	5	15

Suponer que se cuenta con un único procesador y este comienza vacío. Un cambio de contexto insume *una unidad* de tiempo y sólo debe computarse cuando se pasa de procesador vacío a un proceso, cuando se cambia entre procesos distintos o cuando un proceso deja el procesador de nuevo vacío.

a) Realizar un diagrama de Gantt para una política de administración de procesador de tipo *round-robin* con un *quantum* de 5 unidades de tiempo (el cambio de contexto no está incluido en este *quantum*).

b) Realizar un diagrama de Gantt para una política de tipo *shortest remaining time first*.

c) Calcular el tiempo de *turnaround* promedio en ambos casos.

d) A pesar de que una de las dos políticas tiene un tiempo de *turnaround* promedio mucho menor, explicar por qué en algunos contextos podría tener sentido utilizar la otra política. ¿Qué métrica se debe priorizar en dichos contextos?

Parte 2 – Otras políticas de *scheduling*

Ejercicio 12

Suponer que una política de *scheduling* favorece a aquellos procesos que han usado la menor cantidad de tiempo de procesador en el pasado reciente. ¿Por qué se favorecería a los intensivos en E/S, pero a la vez no dejaría a los intensivos en CPU en *starvation*?

Ejercicio 13

Discutir si tendría sentido o no implementar un algoritmo de *scheduling round-robin* que no fuera *preemptive*.

Ejercicio 14 ★

Un sistema que atiende tareas *interactivas* de varias sucursales bancarias está conectado en forma directa a la central policial y, frente a un caso de robo, genera un proceso que activa una alarma en la central.

a) Diseñar un algoritmo que permita que, una vez generado ese proceso de alarma, tenga prioridad sobre el resto de las tareas (recordar que pueden generarse distintas alarmas desde distintas sucursales).
b) Dibujar el diagrama de transición de estados.

Nota: Especificar claramente la forma de administración de las colas.

Ejercicio 15 ★

Consideremos una entidad bancaria que atiende a sus clientes a través de cajeros automáticos y en las ventanillas de sus sucursales. Los cajeros automáticos y las terminales de las ventanillas están conectadas a un sistema central, en donde además se procesan tareas de tipo *batch* que utilizan cintas magnéticas, impresoras y discos. Se desea priorizar por sobre todas las tareas a las tareas *batch*, luego con menor prioridad a los cajeros automáticos y por último las tareas de las terminales.

a) Diseñar una política de administración del procesador que logre este cometido y provea un balance equitativo de los recursos.
b) Indicar la política de administración de cada cola de procesos listos.

Ejercicio 16 ★

Se tiene un sistema donde hay trabajos interactivos y de procesamiento de datos. Los de procesamiento de datos leen archivos inmensos, hacen pequeñas cuentas y los vuelven a grabar.

Se desea que los usuarios interactivos tengan la sensación de buen tiempo de respuesta, pero sin perjudicar excesivamente el *throughput* del sistema.

El *scheduler* puede funcionar con *round-robin* o con FCFS. ¿Qué política utilizaría y por qué? Justificar especialmente por qué la política elegida permite cumplir con ambos objetivos del sistema.

Ejercicio 17 ★

Se tiene un sistema con tres núcleos que puede ejecutar las siguientes tres políticas de planificación (*scheduling*):

- SJF (*Shortest job first*).
- RR1 (*Round-robin* 1), con una cola de procesos por cada núcleo.
- RR2 (*Round-robin* 2), con una única cola global, permitiendo así la migración de los procesos entre los núcleos.

Para el lote de procesos presentado en la siguiente tabla, se pide:

a) Realizar un gráfico de Gantt para cada una de las políticas de planificación.
b) Calcular el *waiting time* y el *turnaround* promedios para cada una de las políticas de planificación.

Proceso	Tiempo de procesamiento (incluye tiempo de bloqueo)	Instante de llegada	Tiempo de bloqueo (período en el cual está bloqueado, inclusive)
P_1	33	2	5-20
P_2	43	4	8-28
P_3	13	14	(no se bloquea)
P_4	23	5	2-12
P_5	8	3	(no se bloquea)
P_6	5	12	(no se bloquea)

Tener en cuenta que: a) el *quantum* es de 5 unidades de tiempo, b) el costo del cambio de contexto es de una unidad de tiempo y c) el costo de migración es de dos unidades de tiempo.