

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.

t	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

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

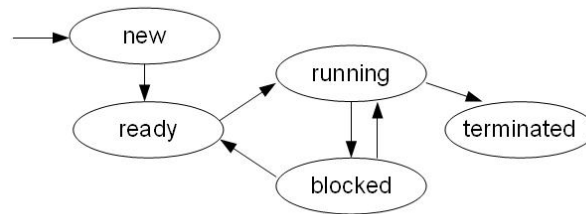
Ejercicio 2 ★

Sean P0, P1 y P2 tales que

- P0 tiene ráfagas cortas de E/S a ciertos dispositivos.
 - P1 frecuentemente se bloquea leyendo de la red.
 - P2 tiene ráfagas prolongadas de alto consumo de CPU y luego de escritura a disco.
- Al planificar estos procesos, ¿cuánto quatum (mucho/poco) y qué prioridad (alta/baja) les asignaría y por qué ?
 - Indique 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) Trabajo más corto primero
- d) FIFO

Ejercicio 5

Considere 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) Dé ventajas y desventajas del esquema del punto a).
- c) ¿Se le ocurre alguna otra modificación para mantener las ventajas sin tener que duplicar punteros?

Ejercicio 6 ★

Considere un sistema en el que hay 10 procesos que son intensivos en el uso de E/S y uno de CPU. Los primeros realizan una operación de E/S cada milisegundo y cada llamado tarda 10 milisegundos en completarse. El cambio de contexto (*context switching*) tarda 0.1 milisegundos y las tareas se ejecutan por muchísimo tiempo.Cuál es la utilización de la CPU para un *scheduling* round robin con:

- a) Quantum de 1 milisegundo
- b) Quantum de 10 milisegundos

Ejercicio 7

Considere el siguiente conjunto de procesos:

Proceso	Tiempo de ráfaga	Prioridad
P1	10	3
P2	1	1
P3	2	3
P4	1	4
P5	5	2

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

- Dibuje 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 tiempo = 1).
- ¿Cuál es el tiempo de espera y de ejecución de cada proceso para cada algoritmo de planificación?
- ¿Cuál de los algoritmos obtiene el menor tiempo medio de espera?

Ejercicio 8

Suponga que los siguientes procesos llegan en los tiempos indicados.

Proceso	Llegada	Procesamiento
P1	0.0	8
P2	0.4	4
P3	1.0	1

- ¿Cuál es el tiempo de *turnaround*^a promedio para estos procesos usando FIFO?
- ¿Cuál es usando Más-corto-primero?
- Más-corto-primero 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 Más-corto-primero.

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

Ejercicio 9

Considere la siguiente tabla de procesos:

Proceso	Procesamiento	Llegada	Prioridad
P1	50 ms	0 ms	4
P2	20 ms	20 ms	1
P3	100 ms	40 ms	3
P4	40 ms	60 ms	2

- Mostrar la planificación de estos procesos utilizando SRT^a, esquema de prioridad sin desalojo y Round-Robin con cuanto de 30 ms.
- ¿Cuál es el tiempo promedio de espera para cada uno de los algoritmos?

^aShortest Remaining Time. Le da prioridad a los procesos cuyo tiempo de corrida restante es menor.

Ejercicio 10 ★

Para los procesos presentados en la siguiente tabla, realizar un gráfico de Gantt para cada una de las políticas 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
P1	1	5
P2	10	6
P3	1	7
P4	10	8

Calcular el *waiting time* y el *turnaround* promedios para cada una de las políticas.

Ejercicio 11

Considere los siguientes procesos:

Proceso	Tiempo de ejecución total	Tiempo de arribo
P1	8	0
P2	8	5
P3	6	14
P4	5	15

Suponer que se cuenta con un único procesador y éste 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.

1. Realice 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*).
2. Realice un diagrama de Gantt para una política de tipo *shortest remaining time first*.
3. Calcule el tiempo de *turnaround* promedio en ambos casos.
4. A pesar de que una de las dos políticas tiene un tiempo de *turnaround* promedio mucho menor, explique 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

Suponga 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, frente a un caso de robo genera un proceso que activa una alarma en la central policial.

- a) Diseñe un algoritmo que permita 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) Dibuje el diagrama de transición de estados.

NOTA: Especifique claramente la forma de administración de las colas.

Ejercicio 15

Piense en 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.

- Diseñe una política de administración del procesador que logre este cometido y provea un balance equitativo de los recursos.
- Indique la política de administración de cada cola de 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? Justifique especialmente por qué la política elegida permite cumplir con ambos objetivos del sistema.

Ejercicio 17 ★

Se tiene un sistema con 3 núcleos que puede ejecutar tres tipos de políticas de planificación (*scheduling*) distintas. Se pide:

- Para el lote de procesos presentado en la siguiente tabla, realizar un gráfico de Gantt para cada una de las políticas de planificación indicadas a continuación.
- Calcular el *waiting time* y el *turnaround* promedios para cada una de las políticas de planificación de procesos.

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.

Lote de tareas:

Proceso	Tiempo de Procesamiento (incluye tiempo de bloqueo)	Instante de llegada	Tiempo de bloqueo (período en el cual está bloqueado, inclusive)
P1	33	2	5-20
P2	43	4	8-28
P3	13	14	(no se bloquea)
P4	23	5	2-12
P5	8	3	(no se bloquea)
P6	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) que el costo de migración es de 2 unidades de tiempo.