



Práctico 3

Planificación de Procesos

- 3.1.** ¿A qué se le llama comúnmente planificador de procesos o *process scheduler*? ¿Cuál es el objetivo que persigue? ¿Cuál es su relación con el término multitarea (*multitasking*)?
- 3.2.** Describir las características de los procesos limitados por I/O y CPU (*I/O-bound* y *CPU-bound*). ¿Dicha clasificación es mutuamente excluyente? ¿Por qué es importante para el planificador distinguir entre estos dos tipos?
- 3.3.** ¿Cuáles son las diferencias entre un esquema de planificación con apropiación y uno cooperativo? Enumerar los principales problemas asociados a cada modelo.
- 3.4.** ¿Por qué los hilos a nivel de usuario generalmente usan planificación cooperativa?
- 3.5.** Definir tiempo de retorno, tiempo de espera y tiempo de respuesta.
- 3.6.** Enumerar ventajas y desventajas de los algoritmos de planificación *First-Come First-Served (FIFO)*, *Shortest-Job-First (SJF)*, *Shortest Remaining Time First (SRTF)*, *Priority* y *Round-Robin (RR)*.
- 3.7.** Indicar cuáles de los siguientes algoritmos podrían producir inanición:
- FCFS
 - SJF
 - Prioridad
 - Round Robin
- 3.8.** Considerando un sistema que utiliza un esquema de planificación *Round-Robin (RR)*, en el cual q es el *quantum*, discutir los siguientes escenarios:
- ¿Qué sucede si q es demasiado grande?
 - ¿Qué sucede si q es demasiado chico?
 - ¿Qué sucede si el tiempo requerido para realizar cambios de contextos es igual a q ?
 - ¿Qué sucede si el tiempo requerido para realizar cambios de contextos es mayor a q ?
 - ¿Qué sucede si el tiempo promedio en el cual se ejecuta un proceso antes de realizar un requerimiento de I/O es mayor a q ?
 - ¿Cuál sería un tamaño de q razonable?

3.9. Considerar el siguiente conjunto de procesos, los cuales arriban en tiempo 0, en el orden dado y con la longitud de ráfaga de CPU dada en milisegundos:

Proceso	Ráfaga
P ₁	10
P ₂	29
P ₃	3
P ₄	7
P ₅	12

Calcular para los algoritmos de planificación FCFS, SJF y RR (*quantum* = 10 milisegundos), cuál resulta en el menor tiempo de espera promedio (*average waiting time*).

3.10. Considerar el siguiente conjunto de procesos, los cuales arriban en tiempo 0, en el orden dado y con la longitud de ráfaga de CPU dada en milisegundos:

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

- Realizar los diagramas de Gantt que ilustran la ejecución de estos procesos utilizando los algoritmos de planificación:
 - FCFS
 - SJF
 - Prioridad no apropiativa (un número de prioridad menor implica una prioridad más alta)
 - RR (*quantum* = 1)
- Calcular el tiempo de retorno (*turnaround time*) de cada proceso para cada algoritmo.
- Calcular el tiempo de espera (*waiting time*) de cada proceso para cada algoritmo.
- ¿Qué algoritmo de planificación resulta en el mínimo tiempo de espera promedio (*minimal average waiting time*)?

3.11. Para el siguiente conjunto de procesos:

Proceso	Arribo	Ráfaga
P ₁	0.000	3
P ₂	1.001	6
P ₃	4.001	4
P ₄	6.001	2

- Realizar los diagramas de Gantt que ilustran la ejecución de estos procesos utilizando los algoritmos de planificación:
 - FCFS
 - SJF
 - Shortest-remaining-time
 - RR (quantum = 2)
 - RR (quantum = 1)
- Calcular el tiempo de retorno promedio (*average turnaround time*) para cada proceso.
- Calcular el tiempo de espera (*waiting time*) para cada proceso.
- ¿Cuál es el *average throughput* en cada caso?

3.12. Para el siguiente conjunto de procesos:

Proceso	Arribo	Ráfaga	Prioridad
P ₁	0.000	4	3
P ₂	1.001	3	4
P ₃	2.001	3	6
P ₄	3.001	5	5

- Mostrar la ejecución de los mismos utilizando diagramas de Gantt para un algoritmo de planificación por prioridades:
 - Apropiativo
 - No apropiativo
- Calcular el tiempo de retorno para cada proceso en ambos casos.
- ¿Cuál es el *average throughput* en cada caso?

3.13. ¿Cuál es la diferencia entre un algoritmo de planificación basado en colas multinivel y uno de planificación de colas multinivel realimentadas?

3.14. Un sistema dado utiliza un esquema de planificación basado en colas multinivel realimentadas. Un determinado proceso, limitado por CPU, requiere de 40 segundos para completar su ejecución. Si la primer cola utiliza un quantum de dos segundos, y en cada nivel se incrementa la fracción de tiempo asignada a cada proceso en 5 unidades, determinar:

- a) La cantidad de veces que el proceso será interrumpido.
- b) La cola en la que finalizará su ejecución.

3.15. Considerar un sistema que utiliza un algoritmo de planificación basado en colas multinivel. ¿Qué estrategia podría ser utilizada a nivel de proceso con el objetivo de maximizar la cantidad de tiempo de CPU asignado al mismo?

3.16. El Planificador tradicional de UNIX impone una relación inversa entre los números de prioridad y las prioridades, es decir que cuanto mayor sea el número, más baja es la prioridad. El planificador recalcula las prioridades de los procesos una vez por segundo usando la siguiente función:

$$Priority = (Uso reciente de la CPU / 2) + base,$$

dónde base = 60 y el uso reciente de la CPU se refiere a un valor que indica la frecuencia con la que un proceso ha utilizado la CPU a partir del cálculo de las últimas prioridades.

Supongamos que el uso reciente de CPU para el proceso P1 es 40, para el proceso P2 es 18, y para el proceso P3 es 10. ¿Cuáles serán las nuevas prioridades de estos tres procesos cuando se recalculen las prioridades? En base a esta información, ¿el planificador tradicional de UNIX aumenta o disminuye la prioridad relativa de un proceso limitado por CPU?

3.17. ¿A qué se refiere la problemática conocida como inversión de prioridades? ¿Por qué es particularmente relevante en el contexto de una planificación de tiempo real?

3.18. ¿Qué entendés por *processor affinity*? ¿Qué relación posee con el concepto de migración en un sistema multiprocesador?

3.19. ¿Qué entendés por balance de carga? ¿Cómo podría implementarse en un sistema multiprocesador? ¿Posee alguna relación con el concepto de afinidad?