



Instituto Tecnológico de Pabellón de Arteaga

Araceli Lucero Cardona

“técnicas de administrador del planificador”

M.S.C Eduardo Flores Gallegos

14-marzo-2018

Planificación Primero en Llegar - Primero en Salir (FIFO).

Los procedimientos son despachados de acuerdo al orden de llegada a la cola de listos. Una vez que un proceso tiene el CPU, se ejecuta hasta su terminación. Esta planificación es No apropiativa; es justa en el sentido formal, pero algo injusta porque los grandes procesos hacen esperar a trabajos pequeños y, los trabajos sin importancia hacen esperar a los trabajos importantes.

La Planificación FIFO ofrece una varianza en tiempo de respuesta relativamente pequeña y es, por tanto, más predecible que otros esquemas; no es un esquema útil en la planificación de procesos interactivos porque no garantiza buenos tiempos de respuesta. También se puede implementar mediante la utilización de una lista. Se reemplazan las páginas de la cabeza y se agregan al final.

Una vez que el proceso obtiene la cpu, se ejecuta hasta terminar, ya que es una disciplina “no apropiativa”.

Puede ocasionar que procesos largos hagan esperar a procesos cortos y que procesos no importantes hagan esperar a procesos importantes.

Es más predecible que otros esquemas.

No puede garantizar buenos tiempos de respuesta interactivos.

Suele utilizarse integrado a otros esquemas, por ejemplo, de la siguiente manera:

- Los procesos se despachan con algún esquema de prioridad.
- Los procesos con igual prioridad se despachan “FIFO”.

Planificación por Prioridad al más corto (SJF, *Short Job First*).

La disciplina del trabajo más corto primero es NO apropiativa y en ella el trabajo o proceso con tiempo estimado de ejecución hasta terminación más corto, es el siguiente en ser ejecutado. El SJF reduce el tiempo de espera de los procesos, sin embargo, tiene una varianza mayor (es decir, es menos predecible) que en FIFO, sobre todo para los trabajos largos.

SJF favorece a los procesos cortos a costa de los procesos largos. Además, selecciona los trabajos que serán atendidos y que dejarán el sistema lo antes posible. Esto último traduce en listas de espera cortas. El SJF es NO apropiativo por lo que resulta de poca utilidad en ambientes de tiempo compartido.

- Es una disciplina no apropiativa y por lo tanto no recomendable en ambientes de tiempo compartido.
- El proceso en espera con el menor tiempo estimado de ejecución hasta su terminación es el siguiente en ejecutarse.
- Los tiempos promedio de espera son menores que con "FIFO".
- Los tiempos de espera son menos predecibles que en "FIFO".
- Favorece a los procesos cortos en detrimento de los largos.
- Tiende a reducir el número de procesos en espera y el número de procesos que esperan detrás de procesos largos.
- Requiere un conocimiento preciso del tiempo de ejecución de un proceso, lo que generalmente se desconoce.
- Se pueden estimar los tiempos en base a series de valores anteriores.

Planificación Round Robin (RR)

Los procesos son despachados en FIFO, pero, se les otorga una cantidad limitada de tiempo de CPU llamada división de tiempo (time - slice) o cuanto (quantum). Si un proceso no termina antes de que se termine su tiempo de CPU, el CPU es apropiado y asignado al siguiente proceso en espera. El proceso apropiado se coloca al final de la lista de listos.

Planeación round robin.

El esquema *Round robin* es efectivo en un ambiente de tiempo compartido en el cual el sistema necesita garantizar un tiempo de respuesta razonable para los usuarios interactivos. La sobre carga de la apropiación se mantiene baja mediante eficientes mecanismos de cambio de contexto y proporcionado suficiente memoria para que los procesos residan en ella al mismo tiempo.

Existe una variante de este esquema llamada *selfish round robin (SRR)*. En este esquema los procesos que entran al sistema se colocan primero en una lista de espera hasta que su prioridad alcanza el nivel de proceso para la lista de activos. Mientras un proceso está en la lista de espera, su prioridad aumenta en una relación a ; cuando un proceso entra a la lista de activos su prioridad se incrementa en una relación b .

Tamaño del cuanto.

La determinación del tamaño del cuanto es decisiva para la operación efectiva de un sistema computacional. ¿Debe ser pequeño o grande el cuanto? ¿Debe ser fijo o variable? ¿Debe ser el mismo para todos los usuarios, o debe ser diferente para cada grupo de usuarios?.

Cuando se tiene un cuanto grande cada proceso puede recibir todo el tiempo que necesita para su terminación, de manera que el esquema round robin se convierte en un FIFO. Cuando el cuanto es pequeño, la sobrecarga por el intercambio de contexto se convierte en un factor dominante y el rendimiento del sistema se degrada.

¿Cuál es el cuanto óptimo ? Es claro que cambia de un sistema a otro y que varía de acuerdo a la carga del sistema.

Queves multi-level

Planificación de Dos Niveles

Los esquemas analizados hasta ahora suponen que todos los procesos ejecutables están en la memoria principal.

Si la memoria principal es insuficiente, ocurrirá lo siguiente

Habrán procesos ejecutables que se mantengan en disco.

- Habrá importantes implicaciones para la planificación, tales como las siguientes:
 - El tiempo de alternancia entre procesos para traer y procesar un proceso del disco es considerablemente mayor que el tiempo para un proceso que ya está en la memoria principal.
 - Es más eficiente el intercambio de los procesos con un planificador de dos niveles.

El esquema operativo de un planificador de dos niveles es como sigue:

1. Se carga en la memoria principal cierto subconjunto de los procesos ejecutables.
2. El planificador se restringe a ellos durante cierto tiempo.
3. Periódicamente se llama a un planificador de nivel superior para efectuar las siguientes tareas:
 1. Eliminar de la memoria los procesos que hayan permanecido en ella el tiempo suficiente.
 2. Cargar a memoria los procesos que hayan estado en disco demasiado tiempo.
4. El planificador de nivel inferior se restringe de nuevo a los procesos ejecutables que se encuentren en la memoria.
5. El planificador de nivel superior se encarga de desplazar los procesos de memoria a disco y viceversa.

Los criterios que podría utilizar el planificador de nivel superior para tomar sus decisiones son los que se indican a continuación:

- ¿Cuánto tiempo ha transcurrido desde el último intercambio del proceso?.
- ¿Cuánto tiempo de cpu ha utilizado recientemente el proceso?.
- ¿Qué tan grande es el proceso? (generalmente los procesos pequeños no causan tantos problemas en este sentido).
- ¿Qué tan alta es la prioridad del proceso?.

El planificador de nivel superior podría utilizar cualquiera de los métodos de planificación analizados.

Multi-level feedback queues.

Colas de Retroalimentación de Niveles Múltiples

Cuando un proceso obtiene la CPU, sobre todo cuando todavía no ha tenido oportunidad de establecer un patrón de comportamiento, el *planificador* no tiene idea de la cantidad de tiempo de CPU que necesitará el proceso. Los procesos limitados por la E/S normalmente usan la CPU sólo un momento antes de generar una solicitud de E/S; los procesos limitados por la CPU pueden usar el procesador durante horas si está disponible en forma *no apropiativa*.

Un mecanismo de planificación debe:

- Favorecer a los trabajos cortos.
- Favorecer a los trabajos limitados por la E/S para lograr un mejor aprovechamiento de los dispositivos de E/S.
- Determinar la naturaleza de un trabajo lo más pronto posible y planificarlo de acuerdo con su naturaleza.

