

Planificación de Procesos y Diagramas de Ocupación del Procesador

Índice de contenidos

Concepto de planificación	2
Objetivos	2
Planificación apropiativa frente a no apropiativa	3
Algoritmos de planificación	4
Utilización de prioridades	4
Utilización de un quantum de tiempo	4
Tipos de algoritmos	4
Que no usan quantum	4
Que usan quantum	6
Diagramas de ocupación del procesador	9

Concepto de planificación

La planificación consiste en un conjunto de políticas y mecanismos incorporados al Sistema Operativo y por los que se rige el orden en que se completa el trabajo que hay que realizar. Su objetivo primordial es optimizar el rendimiento del sistema.

La planificación del procesador es la elección de los procesos que van a ser asignados a los procesadores físicos. La parte del Sistema Operativo encargada de tomar esta decisión se denomina planificador y el algoritmo que utiliza se llama algoritmo de planificación o disciplina.

Objetivos

Para juzgar si un algoritmo de planificación es bueno, se debe tener en cuenta en que medida cumple los siguientes objetivos:

- A. Ser justo: que todos los procesos sean tratados por igual y ninguno pueda experimentar postergación indefinida;
- B. Maximizar la capacidad de ejecución: intentar dar servicio al máximo número posible de procesos por unidad de tiempo;
- C. Maximizar el número de usuarios interactivos recibiendo unos tiempos de respuesta aceptables;
- D. Equilibrar el uso de recursos: para ello los procesos que utilizan recursos infrautilizados deben ser favorecidos;
- E. Lograr un equilibrio entre tiempo de respuesta y utilización de recursos: dependiendo de las características del sistema se debe elegir que factor de estos tiene más importancia;
- F. Evitar la postergación indefinida: se logra mediante el envejecimiento que consiste en que mientras que un proceso espera por un recurso, su prioridad debe aumentar llegando a ser tan alta que el recurso será asignado al proceso;
- G. Asegurar las prioridades: la disciplina debe favorecer a los procesos con prioridades más altas;
- H. Dar preferencia a los procesos que mantienen recursos no compartidos: si un proceso de baja prioridad mantiene un recurso no compartido, este recurso puede ser requerido por procesos de más alta prioridad, por lo cual la disciplina debe otorgar al primer proceso un mejor tratamiento para que libere el recurso lo más rápido posible.

Planificación apropiativa frente a no apropiativa

Una disciplina de planificación es no apropiativa si una vez que se le ha asignado la CPU a un proceso, ya no se le puede retirar. Por lo tanto, será necesario esperar a que el proceso abandone voluntariamente la CPU para poderle asignar a otro proceso.

- Ventajas:
 - Es más fácil predecir los tiempos de ejecución.
- Inconvenientes:
 - Los trabajos largos (que requieren mucho tiempo de CPU) hacen esperar a los cortos.

Una disciplina de planificación es apropiativa si permite que se le pueda retirar la CPU a un proceso mientras que la está usando.

- Ventajas:
 - Útiles en sistemas cuyos procesos de alta prioridad requieren atención rápida.
 - Permiten obtener tiempos de ejecución aceptables.
- Inconvenientes:
 - Conllevan un coste debido a los cambios de procesos (conmutación de la CPU).
 - Requieren sobrecarga debido a la necesidad de mantener en memoria principal programas que no están usando la CPU en todo momento.

Algoritmos de planificación

Mecanismos que utilizan algunos algoritmos de planificación:

Utilización de prioridades

Prioridad: calificador designado a un proceso que determina la atención que va a recibir del sistema.

Según el **agente** que asigna la prioridad, éstas pueden ser:

- Prioridades internas: asignadas automáticamente por el Sistema Operativo;
- Prioridades externas: asignadas desde el exterior.

Según los **cambios** que sufre la prioridad, éstas pueden ser:

- Prioridades estáticas: no cambian durante la ejecución del proceso, es decir no responden a cambios que pueden hacer deseable ajustar alguna prioridad. Los mecanismos de prioridad estática son fáciles de implementar y tienen una sobrecarga relativamente pequeña.
- Prioridades dinámicas: varían durante la ejecución del proceso, por lo que responden a los cambios. Los mecanismos de prioridades dinámicas son más complicados de implementar y suponen una sobrecarga mayor.

Utilización de un quantum de tiempo

Quantum: tiempo máximo de estancia de un proceso en activo.

Ese quantum de tiempo puede ser fijo o variable y puede tener o no el mismo valor para todos los procesos.

Tipos de algoritmos

Teniendo en cuenta si el algoritmo usa o no quantum, estos se dividen en dos grupos:

Que no usan quantum

Dentro de este grupo y ordenados por simplicidad, se describen los siguientes:
Planificación por orden de llegada (FIFO – First Input, First Output, primero en entrar-primero en salir) también conocido como primero en llegar, primero en ser servido (**FCFS, First Come, First Served**).

La lista de procesos en estado listo está ordenada según el orden de llegada al sistema (creación del proceso), siendo el más antiguo el que está en la cabecera de dicha lista. Por

lo tanto, cuando se crea un proceso éste se introduce al final de la lista de procesos en estado listo, y el criterio para elegir el proceso que deberá pasar a estado activo es el primero que está en dicha lista.

Es una disciplina no apropiativa, por lo tanto al proceso que está activo no se le puede retirar la CPU hasta que él la abandone voluntariamente.

- Inconvenientes:
 - Bajos rendimientos, procesos largos hacen esperar a procesos cortos;
 - No es útil en sistemas interactivos puesto que no garantiza buenos tiempos de respuesta.
- Ventajas:
 - Permite predecir el orden de ejecución de los procesos;
 - Es fácil de implementar.
- Variantes:
 - Considerando el tiempo que el proceso lleva en estado listo en vez del tiempo que lleva en el sistema, por lo que se selecciona, para asignar el procesador, el proceso que ha estado más tiempo en la lista de procesos en estado listo.
 - Considerando que la disciplina es apropiativa, esto es que a un proceso se le puede retirar la CPU y pasar de activo a listo, si llega a la lista de procesos en estado listo un proceso más antiguo que él al que se le asignará la CPU.

Planificación del trabajo más corto primero (Shortest Job First o SJF)

Es una disciplina no apropiativa. La lista de procesos en estado listo está ordenada por la cantidad de tiempo de ejecución estimada, de forma que el proceso más corto es el primero en la cola. Para establecer este orden se necesita saber el periodo de tiempo de ejecución de un proceso, y como esto raramente está disponible, se confía en las estimaciones de los usuarios, valor que se almacena en el PCB del proceso.

- Inconvenientes:
 - Con respecto al algoritmo FIFO, es menos predecible los tiempos de respuesta, sobre todo de los procesos largos;
 - Es poco útil en sistemas interactivos puesto que no garantiza tiempos de respuesta razonables.
- Ventajas:
 - Con respecto al algoritmo FIFO, reduce al mínimo el tiempo promedio de espera de los trabajos, ya que al favorecer los trabajos cortos sobre los largos reduce el número de trabajos en espera.

Planificación del tiempo restante más corto (Shortest Remaining Time o SRT)

Esta disciplina elige siempre al proceso que le queda menos tiempo de ejecución estimado para completar su ejecución; de esta forma aunque un proceso requiera mucho tiempo de ejecución, a medida que se va ejecutando irá avanzando en la lista de procesos en estado listo hasta llegar a ser el primero.

Para realizar esta elección, es necesario actualizar el PCB de los procesos a medida que se le asigna tiempo de servicio (activo o bloqueado), lo que supone una mayor sobrecarga adicional.

Es una disciplina apropiativa ya que a un proceso activo se le puede retirar la CPU si llega a la lista de procesos en estado listo otro con un tiempo restante de ejecución estimado menor.

Que usan quantum

También ordenándolos por simplicidad, se describen los siguientes:

Planificación de asignación en rueda, por turnos (Round Robin (RR) Scheduling)

La lista de procesos en estado listo es una cola circular, de forma que el proceso al que se le asigna la CPU pasará posteriormente a ser el último de dicha cola circular. Cuando a un proceso se le asigna la CPU, éste puede usarla durante el tiempo máximo especificado por su quantum. En el caso de que el proceso no agote su quantum debido a que solicita un evento por el cuál debe esperar (pasa a estado bloqueado), entonces se le asignará la CPU al siguiente proceso de la cola circular (proceso que lleva más tiempo en estado listo), pasando a activo sin esperar que se termine el quantum del anterior. En el caso de que el proceso consuma todo su quantum, al pasar de activo a listo, se colocará al final de la lista de procesos en estado listo (de ahí que se considere como una cola circular).

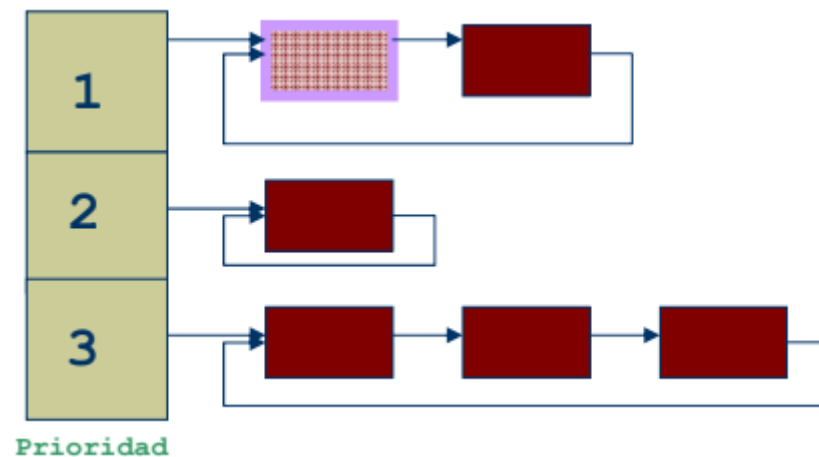
Para insertar un proceso nuevo es indiferente por donde se inserta en la lista de procesos en estado listo.

Esta disciplina apropiativa es efectiva en sistemas de tiempo compartido, puesto que garantiza tiempos de respuesta razonables.

Planificación con colas de niveles múltiples.

A cada proceso se le asigna una prioridad, y la lista de procesos en estado listo está formada por una serie de colas circulares, donde cada una de ellas contiene a todos aquellos procesos que tienen una misma prioridad. La CPU se asigna siempre al Proceso de la cola de mayor prioridad que no esté vacía. Este proceso puede agotar su quantum o no, en ambos casos cuando vuelva a la lista de procesos en estado listo se colocará al final de la cola circular correspondiente a su prioridad.

Supóngase que solo se tienen tres prioridades y que cuanto menor sea el valor de la prioridad mayor será la atención que se le da al proceso. Si se tiene seis procesos dando lugar a la lista de procesos listos que muestra la siguiente figura, el procesador será asignado al primer proceso de la cola de mayor prioridad (proceso con el borde morado), pues dicha cola no está vacía.



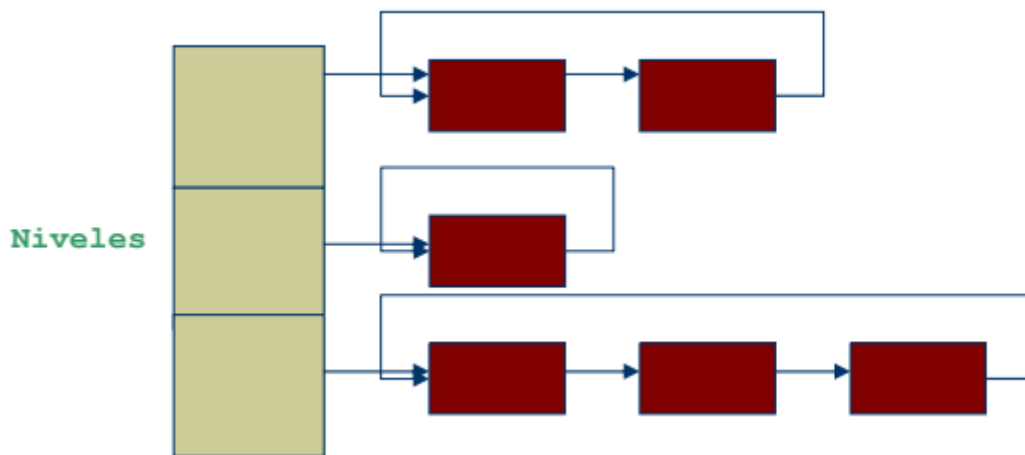
Un problema que presenta esta disciplina es la posibilidad de que los procesos de baja prioridad pueden quedar postergados indefinidamente. Una forma de solucionarlo es con la llamada prioridad por envejecimiento, que consiste en que la prioridad del proceso aumenta gradualmente a medida que éste pasa cierto tiempo sin que se le asigne la CPU, garantizando su terminación en un tiempo finito.

Planificación con colas de retroalimentación de niveles múltiples.

Este algoritmo de planificación pretende tratar a los procesos de acuerdo a su comportamiento (utilización que hace el proceso del procesador), es decir:

- Si el proceso se suspende frecuentemente se le debe asignar bastantes veces el procesador con un quantum pequeño;
- Si el proceso se suspende poco se le debe asignar el procesador menos veces pero con un quantum grande.

Para conseguir esto la lista de procesos en estado listo está formada por varios niveles y cada uno de ellos con una cola circular. Cuando se inserta un nuevo proceso este se añade al final de la cola de mayor nivel moviéndose por dicha cola en orden FIFO hasta obtener la CPU.



La CPU se asigna al proceso que se encuentra en la cabecera de la cola de mayor nivel que no esté vacía. Cuando un proceso recibe la CPU puede ocurrir:

- Que agote su quantum: entonces al pasar a estado listo se coloca al final de la cola del siguiente nivel inferior al que pertenecía. Mientras que el proceso consuma totalmente su quantum en cada nivel, irá moviéndose hacia el final de las colas de los niveles inferiores. La cola del nivel más profundo se gestiona mediante asignación de rueda, cuyos procesos permanecen hasta que terminan.
- Que no agote su quantum: entonces cuando pase de estado bloqueado a listo volverá al final de la cola del nivel al que pertenecía. Se supone que el comportamiento inmediato anterior de un proceso puede ser un buen indicador del comportamiento futuro cercano.

Normalmente, el quantum otorgado al proceso aumenta a medida que el proceso se mueve hacia las colas de los niveles inferiores. De esta forma, cuanto más tiempo estén los procesos en la lista de procesos en estado listo, mayor será su quantum cada vez que obtienen la CPU. En definitiva, este algoritmo trata a los procesos según su comportamiento, y los separa por categorías según sus necesidades de CPU.

Variantes:

- Permitir al proceso moverse hacia un nivel superior al anterior cada vez que abandona voluntariamente la CPU antes de agotar su quantum. De esta forma, se responde a los cambios en el comportamiento de un proceso.
- Mantener al proceso circulando en asignación de rueda varias veces a través de cada cola antes de pasarlo a la siguiente inferior. El número de ciclos en cada uno de los niveles se incrementa a medida que el proceso pasa a la cola del nivel siguiente inferior.

Diagramas de ocupación del procesador

La ocupación de un procesador por parte de los procesos que posee el sistema se puede plasmar mediante un **diagrama de ocupación**. Para la realización de uno de estos diagramas es necesario acordar la nomenclatura que se va a emplear para reflejar los tres posibles estados en los que se pueden encontrar un proceso. Por ejemplo, la siguiente notación:

— Activo == Bloqueado - - - Listo

Por otra parte, cada proceso posee un comportamiento durante su ejecución que se refleja mediante una secuencia de ráfagas de CPU y de Bloqueado (normalmente peticiones de operaciones de Entrada/Salida) del siguiente tipo:

CPU – BLOQ. – CPU – BLOQ. - ... - CPU

La unidad de tiempo empleada para especificar la duración de cada ráfaga es intrascendente, por lo que normalmente se omite. Otros factores que reflejan también dicho comportamiento es el tiempo de llegada del proceso al sistema, tiempo de ejecución del proceso estimado por el usuario y/o la prioridad inicial del proceso.

En un diagrama de ocupación se usa el eje X para representar el tiempo y el eje Y para especificar los procesos existentes en el sistema. A lo largo de todo el tiempo que tardan en ejecutarse estos procesos se debe indicar el estado en que se encuentra cada uno de ellos mediante la nomenclatura acordada, reflejando de esta forma que proceso está haciendo uso en cada momento del procesador o en qué intervalos de tiempo esté ocioso.

Con el fin de aclarar mejor el concepto de diagrama de ocupación de un procesador, se presenta a continuación mediante una tabla la carga de trabajo de un sistema seguido del correspondiente diagrama de ocupación para el algoritmo de planificación FIFO (por orden de llegada al sistema).

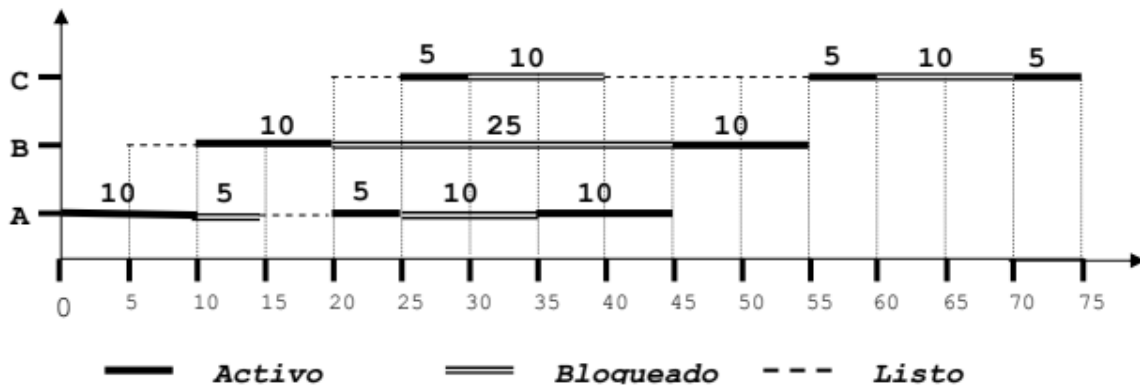
Proceso	Llegada	CPU	BLOQ.	CPU	BLOQ.	CPU
A	0	10	5	5	10	10
B	5	10	25	10	-----	-----
C	20	5	10	5	10	5

Tabla 1: Carga de trabajo del sistema

De está tabla se deduce que en el sistema se crean tres procesos. El significado de cada una de las columnas de dicha tabla es el siguiente:

- **Llegada** representa el momento en el que el proceso llega al sistema;
- **CPU** son las unidades de tiempo de las ráfagas de CPU del proceso;

- **BLOQ.** representan el tiempo que el proceso está bloqueado.



De un diagrama de ocupación del procesador se pueden obtener diversas medidas, entre ellas cabe destacar el **tiempo de ejecución de un proceso** que corresponde al tiempo que dicho proceso ha requerido para completar su ejecución, y que se puede obtener mediante la siguiente operación:

$$\text{Tiempo de ejecución} = \text{Tiempo de finalización} - \text{Tiempo de llegada}$$

(valores referidos al mismo proceso)

Aplicando esta fórmula al ejemplo, se obtienen los siguientes tiempos de ejecución:

$$\text{Tiempo de ejecución (A)} = 45 - 0 = 45$$

$$\text{Tiempo de ejecución (B)} = 55 - 5 = 50$$

$$\text{Tiempo de ejecución (C)} = 75 - 20 = 55$$