

Tema 4. Planificación.

1. Objetivos de la planificación.

La naturaleza o finalidad de cada sistema determinará qué objetivos son los más importantes.

Algunos de los más relevantes son:

• Justicia: Debe de dar una porción justa de ^{tiempo de} CPU a cada proceso. Nadie debe acaparar la CPU o quedar sin servicio.

• Productividad: Cantidad de trabajo desarrollada por unidad de tiempo.

• Tiempo de respuesta aceptable.

• Predecible: Se deben obtener resultados de planificación parecidos en cuanto a métricas.

• Costo extra: El tiempo utilizado en la planificación debe ser el mínimo posible, ya que es tiempo innútil para el sistema. El tiempo empleado por el dispatcher debe ser lo menor posible.

• Ocupación de los recursos: Se debe de maximizar el uso de los recursos del sistema. Se puede dar más prioridad a procesos que requieran recursos poco usados.

• Aplazamiento indefinido: se debe evitar la inanición de los procesos.

• Degradación aceptable: No debe ocurrir una caída global del sistema.

2. Criterios para la planificación.

• Rendimiento o productividad (throughput): Trabajos por unidad de tiempo que completa el sistema. Depende mucho de la longitud de los procesos o trabajos.

• Tiempo de retorno, de residencia, o de estancia: tiempo que transcurre desde el momento en que se crea un trabajo, hasta que se completa. Incluye: tiempo de ejecución efectiva o uso de CPU + el tiempo de espera + tiempos de E/S, etc.

• Tiempo de espera: incluye inicialmente la suma de tiempos de espera hasta que se le concede a un proceso el procesador.

• Tiempo de respuesta o latencia (response time): Tiempo que transcurre desde que se crea un proceso hasta que se obtienen sus primeros resultados. Es una medida buena desde el punto de vista del usuario, al cual le interesa empezar a obtener resultados lo más pronto posible.

• Utilización de la CPU, eficacia, o tiempo de servicio (CPU utilization): Porcentaje de tiempo que está ocupada la CPU de forma útil. Es una métrica importante en grandes sistemas de cómputo, que deben tratar gran cantidad de datos.

3. Modos de decisión

Los algoritmos de planificación se pueden dividir en dos categorías con respecto a la forma en que manejan los instantes de tiempo en que se ejecuta la selección de un proceso: 1) No apropiativos o sin expulsión, 2) Apropiativos o con expulsión.

3.1. No apropiativos.

El planificador selecciona un proceso para ejecutarlo y después deja que se ejecute en CPU hasta que el mismo proceso se bloquea o termina.

Nunca se suspenderá un proceso de manera forzosa, no existe rodaja de tiempo. No se toman decisiones de planificación durante una interrupción. El planificador planificará cuando termine el proceso.

3.2. Apropiativos.

El planificador selecciona un proceso para ejecutarlo y la decisión de expulsarlo de la CPU se toma si suceden alguno de estos ejemplos:

- Se ha ejecutado por un máximo de tiempo fijo. Si sigue en ejecución hasta el final del intervalo, el

proceso se pasa a la cola de Listos y el planificador selecciona otro proceso.

- Llega un nuevo proceso con mayor prioridad.
- Llega una interrupción que pasa un proceso de mayor prioridad de la lista de Bloqueados a la de Listos.

Las políticas expulsivas tienen mayor sobrecarga ya que hay muchos más cambios de contexto, pero pueden proporcionar mejor servicio a la población total de procesos, ya que previenen que cualquier proceso pueda monopolizar el procesador durante mucho tiempo y produzca inanición.

4. Tipos de planificación del procesador

El objetivo de la planificación es asignar procesos a ser ejecutados por el procesador a lo largo del tiempo, de forma que se cumplan los objetivos del sistema tales como el tiempo de respuesta, el rendimiento y la eficiencia del procesador.

En muchos sistemas esta actividad de planificación se divide en tres funciones independientes:

- Planificación a largo plazo que se realiza cuando se crea un proceso, en este punto se decide si se añade al conjunto de los que están activos.
- Planificación a medio plazo, es parte de la función de intercambio o memoria virtual. A veces hay que decidir si añadimos un proceso a memoria principal o a disco.
- Planificación a corto plazo conlleva decidir cual de los procesos ~~Listos~~ para ejecutar es el siguiente a introducir en la CPU.

En términos de frecuencia de ejecución:

- El planificador a largo plazo ejecuta con relativamente poca frecuencia
- El planificador a medio plazo se ejecuta más frecuentemente para tomar decisiones de intercambio.
- El planificador a corto plazo ejecuta mucho más frecuentemente y toma las decisiones de grano fino sobre qué pro-

ceso ejecutar el siguiente.

4.1. Planificación a largo plazo.

El planificador a largo plazo determina qué programas se admiten en el sistema para su procesamiento. Se controla el grado de multiprogramación.

Una vez admitido un programa de usuario, este se convierte en un proceso y se añade a la cola del planificador a corto plazo. En algunos sistemas, si están muy saturados en cuanto al uso de memoria RAM, un proceso de reciente creación comienza cargándose en la memoria swap o virtual, en cuyo caso se añade a la cola de planificador a medio plazo.

Cuanto mayor sea el número de procesos creados y cargados en RAM menor será el porcentaje de tiempo en que cada proceso se pueda ejecutar, es decir, más procesos compiten por la CPU. Así, el planificador a largo plazo puede limitar el grado de multiprogramación a fin de proporcionar un servicio satisfactorio al actual conjunto de procesos cargados en RAM.

Por otro lado, si la fracción de tiempo que el procesador está ocioso excede un determinado valor, se puede invocar al planificador a largo plazo para traer procesos desde estado Nuevo a estado Listo.

La decisión de qué trabajo admitir el siguiente puede basarse en encontrar un compromiso entre procesos limitados por el procesador y procesos limitados por la E/S.

- Se dice que un proceso está limitado por el procesador si realiza mucho trabajo computacional y ocasionalmente usa los dispositivos de E/S.

- Se dice que un proceso está limitado por la E/S si se pasa más tiempo utilizando los dispositivos de E/S que el procesador.

4.2. Planificación a corto plazo.

El planificador a corto plazo se invoca siempre que ocurre un evento que puede conllevar el Bloqueo del proceso actual o su retirada de la CPU a la lista de Listos.

En este caso hay que expulsar el proceso actualmente en ejecución en favor de otro. Téngase como por ejemplo las siguientes circunstancias en las que se puede producir esa expulsión:

- Se produce una interrupción de reloj por rodaja o raga detiempo.
- Se produce una interrupción de E/S
- Se produce un proceso de estado Bloqueado a Listo
- En determinadas llamadas al sistema que provoquen una E/S, como `open()`, `read()`, `close()`.
- Ante determinadas señales o mecanismos de exclusión mutua como los semáforos, los cuales pueden bloquear el proceso actual.
- Cuando un proceso se bloquea a espera de que termine un hijo o un hilo.
- Cuando un proceso finaliza su ejecución, ya que hay que dejarlo de tener en cuenta en la planificación.

En general, cualquier tipo de cambio de estados en un proceso conlleva la invocación del planificador.

4.3. Colas de prioridades multinivel.

La prioridad la establece el sistema inicialmente en función de determinados parámetros como el propietario del proceso, el tipo del procesador, su tamaño, los recursos que requiere, etc.

Si la prioridad de un proceso no cambia durante su ejecución y ésta se usa para planificar a corto plazo, se obtiene una política de prioridades estáticas.

Otro problema de los esquemas de planificación con prioridades estáticas es que los procesos con prioridad más baja pueden sufrir inanición.

Resulta más adecuada una planificación por prioridades dinámicas. Un proceso se crea con una prioridad dinámica que se recalculta periódicamente. La prioridad de los procesos que más tiempo de CPU han gastado disminuye en relación a la prioridad de los procesos que han gastado menos.

4.4. First come first served.

La directiva de planificación más sencilla es FIFO.

En el momento en que un proceso pasa al estado de listo, se une a la cola de Listos. Cuando el proceso actualmente en ejecución deja de ejecutar, se selecciona para ejecutar el proceso que ha estado más tiempo en la cola de listos.

Tenga en cuenta que en este algoritmo de planificación no existen las rodajas de tiempo. Un proceso se puede ejecutar entero o hasta que realice una operación de E/S por la que tenga que esperar.

Proceso	Tiempo de llegada	Tiempo de servicio (T _s)	Tiempo de Comienzo	Tiempo de Finalización	T. de estancia (T _e)	T. de estancia normalizada (T _{en})
W	0	1	0	1	1	1
X	1	100	1	101	100	1
Y	2	1	101	102	100	100
Z	3	100	102	202	199	1.99
					100	26
Media						

El tiempo de estancia normalizado para el proceso Y es excesivamente grande en comparación con los otros procesos. El valor 100 quiere decir que el tiempo total que el proceso está en el sistema es 100 veces mayor que el tiempo requerido para su procesamiento. Sucederá siempre que llegue un proceso corto a continuación de un proceso largo.

Por otra parte, los procesos largos no van mal, por ejemplo Z tiene un tiempo de estancia de casi el doble que Y, pero su tiempo de estancia normalizado está por debajo de 2.0.

FCFS (FIFO)

Proceso	Tiempo de llegada	+ Inicio	Fin	T	E	P
A	0	3	0	3	3	1
B	1	5	3	8	7	2
C	3	2	8	10	7	3.5
D	9	5	10	15	6	1
E	12	5	15	20	8	2
Promedio		4		6.2	2.2	1.74

Otro problema de FCFS es que tiende a favorecer procesos limitados por el procesador sobre los procesos limitados por la E/S. Si un proceso limitado por el procesador llega primero y está ejecutando, el resto de los procesos debe esperar. Algunos de estos estarán en las colas de E/S, pero se pueden mover a la cola de listos mientras que el proceso limitado por el procesador sigue ejecutando. Cuando el proceso actual en ejecución deja el estado Ejecutando, los procesos listos pasarán al estado de Ejecutando y se volverán a Bloquear en un evento de E/S. Si el proceso limitado por el procesador está también Bloqueado, el procesador se quedará ocioso. De esta manera, FCFS puede conllevar VBOS ineficientes del procesador y de los dispositivos de E/S.

FCFS a menudo se combina con esquemas de prioridades para proporcionar una planificación eficaz. De esta forma, el planificador puede mantener varias colas, una por cada nivel de prioridad.

4.5. Turno rotatorio (round robin)

Una forma directa de reducir el castigo que tienen los trabajos cortos con FCFS es la utilización de expulsión basándose en el reloj. La política más sencilla es la planificación cíclica.

Se genera una interrupción de reloj cada cierto intervalo de tiempo. Cuando sucede la interrupción, el proceso actual en ejecución se sitúa en la cola de listos, y se selecciona el siguiente trabajo según la política FCFS. A cada proceso se le da una rodaja antes de ser expulsado.

Con la planificación en turno rotatorio, el tema clave de diseño es la longitud del quantum de tiempo o rodaja a ser utilizada. Si el quantum es muy pequeño, el proceso se moverá por el sistema relativamente rápido. Se deben evitar los quantum de tiempo muy pequeños. Una buena idea es que el quantum de tiempo debe ser ligeramente mayor que el tiempo requerido para una interacción o una función típica del proceso.

En el caso extremo de un quantum de tiempo mayor que el proceso más largo en el sistema, la planificación en turno rotatorio genera en FCFS, pudiendo provocar inanición y favoreciendo a los procesos largos.

Una desventaja de la planificación en turno rotatorio es que trata de forma desigual a los procesos limitados por el procesador y a los limitados por la E/S. Si hay una mezcla de los dos tipos de procesos sucederá lo siguiente:

Un proceso limitado por la E/S utiliza el procesador durante un periodo corto y luego se Bloquea; espera a que se complete la operación de E/S y a continuación se une a la cola de Listos. Por otra parte, un proceso limitado por el procesador generalmente utiliza la rodaja de tiempo completa mientras ejecuta e inmediatamente vuelve a la cola de Listos. Los procesos limitados por la E/S normalmente no aprovecharán la rodaja de tiempo completa, porque antes pasarán a estado Bloqueado a la espera de que termine la operación de E/S que hayan invocado.

Introducimos una cola auxiliar FCFS a la que se mueven los procesos Bloqueados en una E/S.

Cuando se va a tomar una decisión de activación, los procesos de la cola auxiliar tienen prioridad sobre la cola de Listos. Cuando se activa un proceso desde la cola auxiliar, éste ejecuta por un tiempo no superior a la rodaja de tiempo menos el tiempo total que ha estado ejecutando desde que no está en la cola principal de Listos. Algunos estudios de rendimiento indican que este enfoque es realmente superior al turno rotatorio en términos de equidad.

4.6. Primero el más corto(SPN)

SPN es una política no expulsiva en la que se selecciona el proceso con el tiempo de procesamiento más corto esperado, es decir, el que necesite menos CPU o tiempo de servicio para ejecutarse y salir del sistema.

Una vez dentro de la CPU un proceso se ejecuta enteramente sino se bloquea.

Con esta política de planificación es necesario contar con información por anticipado acerca del tiempo de CPU que requieren los procesos que forman la lista de Listos. Normalmente se estima mediante:

- Modelos matemáticos que asignen un tiempo de servicio dependiendo del tipo de proceso y carga del sistema
- La caracterización de las necesidades del proceso, es decir, ver si durante su historia de ejecución ha sido tendiente a manejar ráfagas limitadas por E/S o limitadas por procesador, y cuál es su tendencia actual.

El riesgo con SPN es la posibilidad de inanición para los procesos más largos si hay una llegada constante de procesos más cortos, aunque hay casos en los que se penaliza fuertemente al proceso corto.

4.7. Menor tiempo restante (SRT)

Esta política expulsa procesos sin tener en cuenta su rodaja de tiempo.

Si los procesos que van llegando a la lista de Listos no tienen tiempos restantes menores que el proceso que actualmente está en ejecución éste último no se expulsa, ya que no hay asignación de quantum o rodaja de tiempo.

Se deben almacenar los tiempos de servicio transcurridos, generando sobrecarga, además hay que interrumpir el proceso en ejecución para calcular el tiempo restante de los nuevos procesos que llegan a la cola de Listos.

Las políticas SPN y SRT no se pueden implementar en sistemas en los que no se conoce a priori el tiempo de CPU que requieren los procesos situados en la lista de Listos según su tipo.

4.8. Planificación retroalimentada multinivel (FB)

Otra forma de establecer una preferencia para los trabajos más cortos es penalizar a los que han estado ejecutando más tiempo.

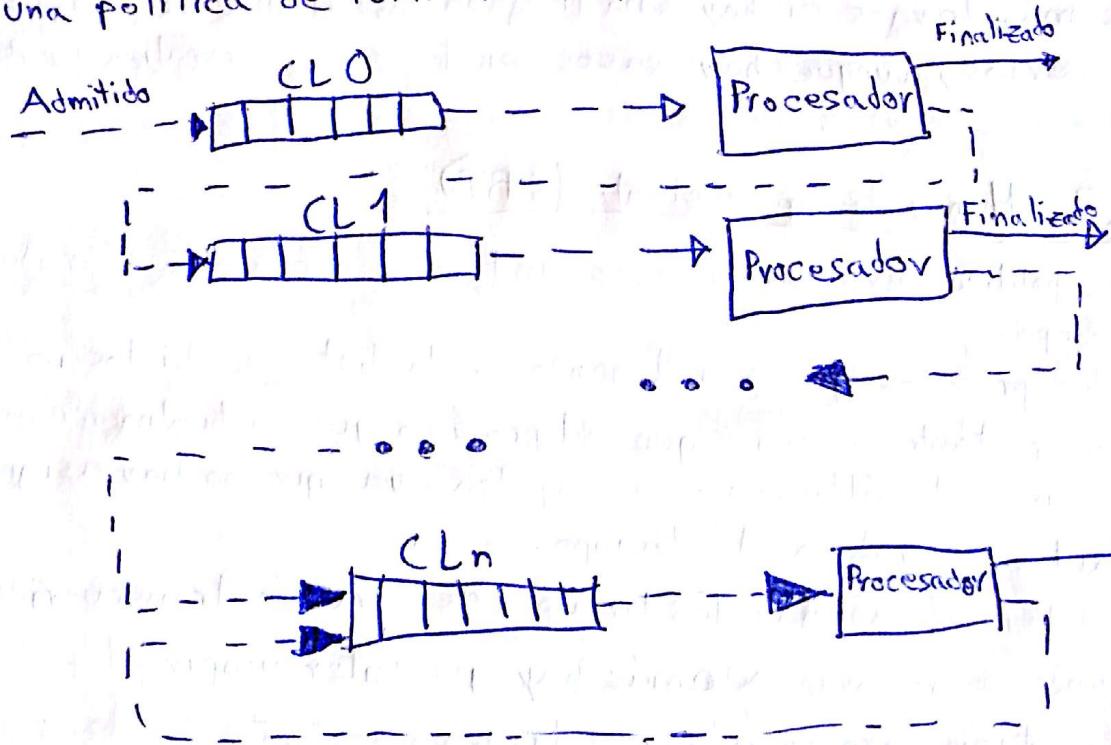
La planificación FB se realiza con expulsión y se utiliza un mecanismo de prioridades dinámica donde existen varias colas

que utilizan un mecanismo FCFS, excepto en la cola de menor prioridad.

Cuando un proceso entra en el sistema, se sitúa en la cola de Listos CL0. Cuando se produce su primera expulsión sale de la cola CL0, y cuando vuelve al estado Listo, se sitúa en CL1.

El planificador asignará al dispatcher el primer proceso que haya en la cola CL0, si no hay procesos se pasa a la cola CL1 y se coge el 1º proceso que haya según FIFO.

Un proceso más largo irá degradándose gradualmente. Así, se favorece a los procesos nuevos más cortos sobre los más viejos y largos. La cola con menor prioridad se trata con una política de turno rotatorio.



4.9. La Planificación contribución justa

En la política FSS se toman decisiones de planificación basándose en conjuntos de procesos.

FSS considera el histórico de ejecución de un grupo de procesos relacionados, junto con el histórico de ejecución de cada uno de los procesos, para tomar decisiones de planificación. El sistema divide a los procesos en un conjunto de grupos y destina una fracción del procesador a cada grupo.

La planificación se realiza en base a la prioridad, con una única cola de Listos, y tiene en cuenta la prioridad del proceso,

su uso reciente del procesador y el uso reciente de procesador del grupo al que pertenece el proceso.

Cuanto mayor sea el valor numérico de la prioridad, menor será la prioridad.

En FSS, la siguiente fórmula se aplica al proceso j del grupo K :

$$CPU_j(i) = \frac{CPU_j(i-1)}{2}$$

$$GCPU_k(i) = \frac{GCPU_k(i-1)}{2}$$

$$P_j(i) = Base_j + \frac{CPU_j(i)}{2} + \frac{GCPU_k(i)}{4 \times W_k}$$

• $CPU_j(i)$ = Medida de utilización del procesador por el proceso j en el intervalo i .

• $GCPU_k(i)$ = Medida de utilización del procesador del grupo K en el intervalo i . Al igual que antes, depende de esa misma medida en un intervalo anterior. La primera vez que un proceso llega al sistema se le asigna un valor de $GCPU$ que tengan los procesos del grupo al que pertenece. O si el grupo lo forma ese único proceso.

• $P_j(i)$ = Prioridad del proceso j al comienzo del intervalo i ; valores más pequeños equivalen a prioridades más altas.

• $Base_j$ = Prioridad base del proceso j , al comienzo del intervalo i ; valores más pequeños corresponden a prioridades más altas. Se asigna por el sistema según su tipo. La primera vez que un proceso llega al sistema se le asigna una prioridad, de forma que $P_j(0) = Base_j$.

• W_k = Prima asignada al grupo K , con la restricción que

$$0 < W_k \leq 1 \text{ y } \sum_k W_k = 1$$

A cada proceso se le asigna una prioridad base. La prioridad de un proceso disminuye a medida que el proceso utiliza el procesador y a medida que el grupo al que pertenece utiliza el procesador. Mientras mayor sea el peso asignado al grupo, su utilización afectará menos a su prioridad.

Tiempo	Proceso A			Proceso B			Proceso C		
	Prioridad	Contador CPU	Contador Grupo	Prioridad	Contador CPU	Contador Grupo	Prioridad	Contador CPU	Contador Grupo
0'	60	0	0	60	0	0	60	0	0
1	90	30	30	60	1	1	60	0	0
2	74	45	15	98	130	30	75	0	30
3	96	37	37	74	45	15	67	0	15
4	78	18	18	81	7	37	93	130	37
5	98	39	39	70	5	18	76	15	18

Grupo 1

Grupo 2

Ejemplo en el que el proceso A está en un grupo y los procesos B y C están en un segundo grupo, donde se asume que:

• Cada grupo tiene una prima de 0'5 = 0

- Todos los procesos están limitados por el procesador y están normalmente listos para ejecutar.
- Todos tienen una prioridad base de 60
- Se asigna una rodaja de tiempo de 1 segundo.
- La utilización del procesador se mide de la siguiente manera: El procesador para la ejecución actual 60 veces por segundo, de forma que en cada interrupción se incre-

menta el campo del uso del procesador del proceso actualmente en ejecución, así como el campo del grupo correspondiente, para posteriormente retomar la ejecución del proceso actual.

• Las prioridades se recalculan una vez por segundo. En las divisiones que se realizan para calcular las prioridades se redondea para obtener prioridades enteras. Observemos la prioridad 74 del proceso A en el instante 2, se calcula así:

$$CPU_{a(2)} = CPU_{a(1)}/2 = 30/15 = 15 ??$$

$$GCPU_{a(2)} = GCPU_{a(1)}/2 = 30/2 = 15$$

$$P_a(2) = \text{base} + CPU_{a(2)}/2 + GCPU_{a(2)} / (4 * 0.5) = 60 + (15/2) + (15/(4 * 0.5))$$

74, cuando debería dar 75 pero como $15/2 = 7.5$, se redondea a 7.

5. Comparación de rendimiento

No es lo mismo un sistema de propósito general que sistemas de cómputo de propósito específico, donde sabemos qué tipos de procesos va a haber en el sistema y para qué se va a utilizar fundamentalmente. En este último caso, la elección de una política de planificación es algo más sencillo y concreto.